Surrogate "Level-Based" Lagrangian Relaxation for Mixed-Integer Linear Programming

Mikhail Bragin

March 17, 2022

Surrogate "Level-Based" Lagrangian Relaxation for Mixed-Integer Linear Programming

Mikhail A. Bragin
Department of Electrical and Computer Engineering, University of Connecticut, mikhail.bragin@uconn.edu

Mixed-Integer Linear Programming problems, playing a prominent role in Operations Research, are prone to the curse of dimensionality; specifically, the combinatorial complexity of the associated MILP problems increases exponentially with the increase of the problem size. To efficiently solve the "separable" MILP combinatorial problems, decomposition, and coordination Surrogate "Level-Based" Lagrangian Relaxation method is developed. The new method efficiently exploits the underlying geometric-convergence potential - the best convergence theoretically possible - inherent to Polyak's step-sizing formula in the dual space without the need to know the optimal dual value required for convergence. Unlike all the previous methods to obtain "level" estimates of the optimal dual value by adjusting hyperparameters, the key novel idea to obtain "level" value is through a hyper-parameter-free optimization: a novel "multiplier-convergence-feasibility" Linear Programming constraint satisfaction problem needs to be solved, from which, in conjunction with the Polyak's formula, the sought-for "level" estimate is inferred. Testing results for medium- and large-scale instances of standard Generalized Assignment Problems from OR-library demonstrate that 1. "level" estimates generally decrease and the dual values increase faster than those obtained by using other existing methods, 2. the new method is computationally efficient in terms of the quality of solutions (for several large-scale cases, optimal solutions are obtained) and in terms of the CPU time, 3. the new method is robust with respect to the choice of initial parameters such as step-sizes and multipliers. Moreover, the procedure to estimate the "level" value is independent of linearity and separability of the problem, and thus the new method also paves the way for efficient resolution of general Mixed-Integer Programming (MIP) problems.

Key words: Combinatorial Optimization; Lagrangian Relaxation; Mixed-Integer Linear Programming

Subject classifications: Programming: Integer (Relaxation-Subgradient);

Area of review: Discrete Optimization;

1. Introduction

We study the resolution of combinatorial optimization problems belonging to a class of "separable" mixed-integer linear programs (MILP) with integer linear programming (ILP) being a special case, which can formally be stated as

$$\min_{(x,y):=\{x_i,y_i\}_{i=1}^{I}} \left\{ \sum_{i=1}^{I} \left((c_i^x)^T x_i + (c_i^y)^T y_i \right) \right\}, \tag{1}$$

$$s.t.\ g(x^k,y^k)=0,\ \{x_i,y_i\}\in\mathcal{F}_i, i=1,\ldots,I. \tag{2}$$

where $g(x^k,y^k)=\sum_{i=1}^{I} A_i^x x_i + \sum_{i=1}^{I} A_i^y y_i - b$. The primal problem (1)-(2) is assumed to be feasible and feasible regions $\mathcal{F}_i \subset \mathbb{Z}^{n_i} \times \mathbb{R}^{n_i}$, which can be delineated by simple bounds, subsystem constraints or both, are assumed to be finite. Mixed-integer linear programming has multiple applications in problems of importance, which will be reviewed in Section 2.

MILP problems are known to belong to the NP-hard class of problems because of the presence of integer variables x. Problems of practical sizes are thus difficult to solve because of the associated combinatorial complexity: as the problem size increases linearly, the complexity increases exponentially. The reduction of complexity, on the other hand, has been traditionally achieved through the use of the dual decomposition and coordination Lagrangian Relaxation technique: decomposition "reverses" the combinatorial complexity thereby exponentially reducing the effort required to solve sub-problems: sub-problem i corresponds to a subsystem i. Notwithstanding the advantage of the decomposition aspect, the coordination aspect of the method, however, has been the subject of intensive research for many decades because of fundamental difficulties of the non-smooth optimization of the associated dual functions as explained ahead.

The purpose of this paper is two-fold: 1. To provide an overview of the important milestones in the development of the optimization of convex non-smooth functions as well as the Lagrangian Relaxation method, the success of which relies on the optimization of non-convex dual function in Section 2, and 2. To develop a solution methodology to solve the problem (1)-(2) through the use of Lagrangian Relaxation (LR) to exploit the efficient reduction of complexity while developing the novel approach to overcome coordination difficulties of previous methods in Section 3.

2. Lagrangian Duality for Discrete Programs and Non-Smooth Optimization

The convex dual function corresponding to the MILP problem (1) is:

$$\max_{\lambda}\{q(\lambda):\lambda\in\mathbb{R}^m\}, \tag{3}$$

where

$$q(\lambda)=\min_{(x,y)}\left\{\sum_{i=1}^{I}\left((c_i^x)^T x_i+(c_i^y)^T y_i\right)+\lambda^T\cdot\left(\sum_{i=1}^{I}A_i^x x_i+\sum_{i=1}^{I}A_i^y y_i-b\right),\{x_i,y_i\}\in\mathcal{F}_i, i=1,\ldots,I\right\}. \tag{4}$$

The Lagrangian multipliers λ ("dual" variables) are the decision variables with respect to the dual problem (3) and it is assumed that the set of optimal solutions is not empty. The minimization within (4) with respect to $\{x,y\}$ is referred to as the "relaxed problem." Because of the presence of integer variables x, the dual function (3) is non-smooth with facets (each corresponding to a particular solution to the relaxed problem within (4)) intersecting at ridges whereby derivatives of $q(\lambda)$ exhibit discontinuities; in particular, the dual function is not differentiable at λ^*. As a result, subgradients (in early literature referred to as "generalized gradients") may lead to lower dual values or exhibit zigzagging across ridges of the dual function (see Figure 1 for illustrations).

Traditionally, the dual function (3) is maximized by updating Lagrangian multipliers λ based on stepsizes s^k and subgradient directions $g(x^k,y^k)$, which within the Lagrangian relaxation framework are defined as levels of constraint violations $g(x^k,y^k)\equiv\left(\sum_{i=1}^{I}A_i^x x_i^k+\sum_{i=1}^{I}A_i^y y_i^k-b\right)$, as

$$\lambda^{k+1}=\lambda^k+s^k\cdot g(x^k,y^k), \tag{5}$$

where $\{x_i^k,y_i^k\}_{i=1}^{I}$ is a an optimal solution to the relaxed problem (4) with multipliers equal to λ^k. If inequality constraints $\sum_{i=1}^{I}A_i^x x_i+\sum_{i=1}^{I}A_i^y y_i\le b$ are present, one way to handle them is to convert into equality constraints by introducing non-negative real-valued slack variables z such that $\sum_{i=1}^{I}A_i^x x_i+\sum_{i=1}^{I}A_i^y y_i+z=b$.

Because of the lack of differentiability of the dual function, notably, at the optimum λ^*, the stepsize selection plays an important role to guarantee convergence to the optimum as well as for the success of the overall Lagrangian Relaxation methodology for solving MILP problems. Accordingly, the roadblocks on the way of Lagrangian relaxation are the following:

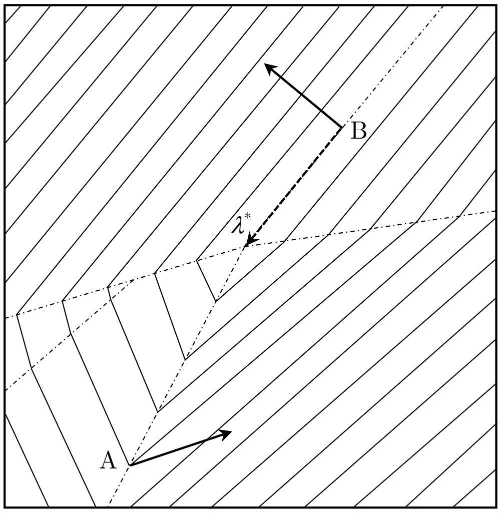

Figure 1 An example of a dual function demonstrating difficulties faced by subgradient methods. Solid lines denote the level curves, dash-dotted lines denote the ridges of the dual function whereby the usual gradients are not defined (possible subgradient directions at points A and B are denoted by solid arrows), and the direction from point B toward optimal multipliers is denoted by a dashed line.

1. Non-differentiability of the dual function

(a) Subgradient directions are non-ascending;

(b) Zigzagging of multipliers across ridges of the dual function leading to many iterations required for convergence;

(c) It is necessary for step-sizes to approach zero to guarantee convergence, yet, it is not sufficient;

2. High computational effort required to solve the relaxed problem in order to obtain subgradient directions;

3. Solutions $\{x_i^k, y_i^k\}$ to the relaxed problem, when put together, do not satisfy constraints (2) even at λ^*.

In the following, by looking through a prism of the above-mentioned difficulties and to lay out the foundation for further methodological developments, several stages in the methodological development to optimize non-smooth dual functions as well as to solve mixed-integer programming problems by using Lagrangian Relaxation will be reviewed roughly in chronological order.

2.1. 1960's: Minimization of "Unsmooth Functionals."

The study of optimization of the class non-smooth convex functions, with $q(\lambda)$ being its member, though irrespective to Lagrangian relaxation, originates in Polyak's seminal work (Polyak (1967)).

Subgradient Method with "Non-Summable" Stepsize. Polyak noted that subgradient directions are not always descending (Polyak 1967, p. 593) for minimization problems (accordingly, subgradient directions are not always ascending for maximization problems such as (3). This property of subgradient as point A is shown in Figure 1.) Nevertheless, convergence to the optimal solution optimizing the non-smooth function was proven under the following (frequently dubbed as "non-summable") stepsizing formula satisfying the following conditions:

$$s^k > 0, \quad \lim_{k \to \infty} s^k = 0, \quad \sum_{k=0}^{\infty} s^k = \infty. \tag{6}$$

A specific example of a "non-summable" step-size is $s^k = \frac{s^0}{k}$, where s^0 is some positive constant.

Subgradient Method with Polyak's Stepsize. As noted by Polyak in his subsequent work (Polyak (1969)), the non-summable stepsizes satysfying (5) lead to "extremely slow convergence" (Polyak 1969, p. 15). Intending to achieve geometric (also referred to as "linear") rate of convergence so that $\|\lambda^k - \lambda^*\|$ is monotonically decreasing, Polyak proposed the stepsizing formula (nowadays bearing his name, or sometimes referred to as "dynamic stepsize rule"), which in terms of the problem under consideration takes the following form:

$$0 < s^k < \gamma \cdot \frac{q(\lambda^*) - q(\lambda^k)}{\|g(x_i^k, y_i^k)\|^2}, \gamma < 2. \tag{7}$$

Within (7) and thereafter in the paper, the standard Euclidean norm will be used. A noteworthy feature of stepsizes (6) and (7) is convergence to zero is required to achieve convergence to λ^*. Even though the optimal dual value $q(\lambda^*)$ is not known, the geometric convergence potential offered by the formula (7) has been extensively exploited and gave rise to applications of Lagrangian relaxation to solve MILP problems as will be discussed next.

2.2. 1970's, 1980's and Early 1990's: Application of Lagrangian Relaxation and Non-Smooth Optimization to Mathematical Programming Problems.

The 1970's and 1980's have been marked by the explosion of the applications of non-smooth optimization with Polyak's step-size (7) for solving a host of discrete (both integer and mixed-integer) optimization problems. Since the optimal dual value $q(\lambda^*)$ of the associated dual value is generally unknown, to compute stepsizes, $q(\lambda^*)$ was estimated, for example, though a "target" value $\overline{q}$ (e.g., Held and Karp (1971)) or by using a feasible cost of the primal problem (e.g., Fisher (1976)). Notable applications of the method are the *traveling salesman problems* (Held and Karp (1970), Held and Karp (1971)), *scheduling problems* (Fisher (1973), Fisher (1976), Muckstadt and Koenig (1977), Shepardson et al. (1980), Hoitomt et al. (1990), Chang et al. (1990), Hoitomt et al. (1993), Yan et al. (1993)), *location problems* (Cornuejols et al. (1977), Erlenkotter (1978)) and many others. An excellent summary of early applications of Lagrangian relaxation for discrete programming problems can be found at Fisher (1981) and Fisher (1985).

2.3. The 1990's: The Subgradient-Level Method.

The subgradient-level method, first proposed by Brännlund (1993) and subsequently improved by Goffin and Kiwiel (1999), overcomes the unavailability of the optimal value needed to compute Polyak's step-size (7) by adaptively adjusting a "level" estimate based on detection of "sufficient descent" of the function and "oscillation" of solutions.

In terms of the problem (3), the procedure of the method is explained as follows: the "level" estimate $q_{lev}^k = q_{rec}^{k_j} + \delta_j$ is used in place of the optimal dual value $q(\lambda^*)$, where q_{rec}^k is the best dual value ("record objective value") obtained up to an iteration k, and δ_j is an adjustable parameter with j denoting the j^{th} update of q_{lev}^k. The main premise behind is when δ_j is "too large," then multipliers will exhibit oscillations while travelling a significant (predefined) distance R without improving the "record" value. In this case, the parameter δ_j is updated as $\delta_{j+1} = \beta \cdot \delta_j$ with $\beta = \frac{1}{2}$. On the other hand, if δ_j is such that the dual value is sufficiently increased: $q(\lambda^k) \geq q_{lev}^k + \tau \cdot \delta_j$, with $\tau = \frac{1}{2}$, then the parameter δ_j is unchanged and the distance travelled by multipliers is reset to 0 to avoid premature reduction of δ_j by β in future iterations.

2.4. Fundamental Difficulties of Sub-gradient Methods.

High Computational Effort. Up to this point in the literature review, the methods reviewed relied on the assumption that non-smooth functions (e.g., the dual function) are given. However, within the Lagrangian relaxation framework, this is not the case since the minimization of the relaxed problem within (4) needs to be performed to obtain a dual function $q(\lambda)$, which is a non-trivial computational task for problems of practical sizes. In fact, even for a given value of multipliers λ^k, minimization within (4) to obtain a dual value $q(\lambda^k)$ and the corresponding subgradient is time-consuming.

Zigzagging of Multipliers. As hypothesized by Goffin (1977), slow convergence of subgradient-based method is due to ill-conditioning. The condition number μ is formally defined as $\mu = \inf\{\mu(\lambda) : \lambda \in \mathbb{R}^m/P\}$, where $P = \{\lambda \in \mathbb{R}^m : q(\lambda) = q(\lambda^*)\}$ (the set of optimal solutions) and $\mu(\lambda) = \min_u \frac{u^T \cdot (\lambda^* - \lambda)}{\|u^T\| \cdot \|\lambda^* - \lambda\|}$ (the cosine of the angle that subgradient form with directions toward the optimal multipliers). It was then demonstrated experimentally when solving, for example, scheduling (Czerwinski and Luh 1994, Fig. 3(b), p. 104) as well as power systems problems (Guan et al. 1995, Fig. 4, p. 774) that multipliers frequently zigzag across the ridges of the dual function. This phenomenon could be attributed to the ill-conditioning of the problems.

To address these difficulties, the notions of "surrogate," "interleaved" and "incremental" subgradients, which do not require relaxed problems to be fully optimized to speed up convergence, emerged in the late 1990's, early 2000's as reviewed next.

2.5. The late 1990's: The Interleaved Subgradient and the Surrogate Subgradient Methods.

Within the Interleaved Subgradient method proposed by Kaskavelis and Caramanis (1998), multipliers are updated after solving one subproblem at a time

$$\min_{(x_i,y_i)} \left\{ (c_i^x)^T x_i + (c_i^y)^T y_i + \lambda^T \cdot (A_i^x x_i + A_i^y y_i), \{x_i, y_i\} \in \mathcal{F}_i \subset \mathbb{Z}^{n_i} \times \mathbb{R}^{n_i} \right\}, \tag{8}$$

rather than solving all the subproblems as within the standard subgradient method. This significantly reduces computational effort, especially for problems with a large number of subsystems.

The more general Surrogate Sub-gradient Method was then developed by Zhao et al. (1999) whereby the exact optimality of the relaxed problem (or even subproblems) is not required. As long as the following "surrogate optimality condition" is satisfied

$$L(\tilde{x}^k, \tilde{y}^k, \lambda^k) < L(\tilde{x}^{k-1}, \tilde{y}^{k-1}, \lambda^k) \tag{9}$$

the multipliers can be updated by using the following formula

$$0 < s^k < \gamma \cdot \frac{q(\lambda^*) - L(\tilde{x}^k, \tilde{y}^k, \lambda^k)}{\left\|g(\tilde{x}^k, \tilde{y}^k)\right\|^2}, \ \ \gamma < 1. \tag{10}$$

and the convergence to λ^* is guaranteed. Here "tilde" is used to distinguish solutions $\{x_i^k, y_i^k\}$, which are obtained by optimally solving the relaxed problem, from the solutions of the relaxed problem $\{\tilde{x}_i^k, \tilde{y}_i^k\}$, not necessarily optimal, but satisfying the "surrogate optimality condition" (9). Unlike that in Polyak's formula, parameter γ is less than 1 to guarantee that $q(\lambda^*) > L(\tilde{x}^k, \tilde{y}^k, \lambda^k)$ so that the step-sizing formula (10) is well-defined in the first place, as proved in (Zhao et al. 1999, Proposition 3.1, p. 703). The concomitant reduction of multiplier zigzagging has been also observed.

2.6. The Early 2000's: Incremental Subgradient Methods.

The main idea of the incremental subgradient method, like that of the interleaved method discussed above, is to improve convergence by solving a subproblem i before updating multipliers. After one subgradient component is updated, rather than updating all the multipliers "at once," within the incremental subgradient methods, multipliers are updated "incrementally." After the i^{th} subgradient component is calculated,

$$\psi_i^k = \psi_{i-1}^k + s^k \cdot \left(A_i^x x_i^k + A_i^y y_i^k - \beta_i\right). \tag{11}$$

Here β_i are the vectors such that $\sum_{i=1}^{I} \beta_i = b$, for example, $\beta_i = \frac{b}{I}$. Only after all i subproblems are solved, are the multipliers "fully" updated as

$$\lambda^{k+1} = \psi_I^k. \tag{12}$$

Convergence results of the subgradient-level method (Goffin and Kiwiel (1999)) have been extended for the subgradient methods and proved. Variations of the method were proposed with β and τ belonging to an interval $[0,1]$ rather than being equal to $\frac{1}{2}$. Moreover, to improve convergence, rather than using constant R, a sequence of R_l was proposed such that $\sum_{l=1}^{\infty} R_l = \infty$.

2.7. 2010's: The Surrogate Lagrangian Relaxation Method.

The Surrogate Lagrangian Relaxation (SLR) method (Bragin et al. (2015)) overcomes the unavailability of the optimal dual value difficulty based on the "contraction mapping" concept. Namely, within the method, distances between multipliers at consecutive iterations are required to decrease, i.e.,

$$\|\lambda^{k+1} - \lambda^{k}\| = \alpha_k \cdot \|\lambda^{k} - \lambda^{k-1}\|, \quad 0 \le \alpha_k < 1. \tag{13}$$

Based on (10), the step-sizing formula has been derived:

$$s^k = \alpha_k \cdot \frac{s^{k-1}\|g(\tilde{x}^{k-1}, \tilde{y}^{k-1})\|}{\|g(\tilde{x}^{k}, \tilde{y}^{k})\|}. \tag{14}$$

Moreover, a specific formula to set α_k has been developed to guarantee convergence:

$$\alpha_k = 1 - \frac{1}{M \cdot k^{1-\frac{1}{k^r}}}, \; M \ge 1, \; 0 \le r \le 1. \tag{15}$$

2.8. The Late 2010's - Early 2020's: Important Applications of Lagrangian Relaxation for MILP Problems.

Lagrangian Relaxation remains a prominent method for solving MILP (sometimes even MINLP) problems within a variety of disciplines with applications to problems of importance to society: building evacuation during fire (Lu et al. (2021)) and active shooter events (Gunn et al. (2017)), energy harvesting-aided cellular IoT (Cheng et al. (2021)), microgrid coordination (Wang et al. (2021)), air traffic flow management (García Heredia (2021)), job-shop scheduling (Yan et al. (2018), Yan et al. (2020)), flow-shop scheduling (Hong et al. (2019)), steelmaking-continuous casting process scheduling (Cui et al. (2020), Cui et al. (2021)), flexible scheduling (Hämmerle and

Weichhart (2017)), plant factory crop scheduling (Huang et al. (2020)), unit commitment (Kim et al. (2018), Chen et al. (2019), Li et al. (2019), Chen et al. (2020), Li et al. (2020)), neural network acceleration (Peng et al. (2021)), semiconductor manufacturing (Chang and Dong (2017)), chiller-plant optimization (Zhang et al. (2019)), slot allocation of a liner container shipping service (Wang et al. (2021)), and transmission and distribution coordination (Bragin and Dvorkin (2021)). The above-mentioned list of applications encompasses research during the last 5 years, and by no means inclusive.

In short summary, all the subgradient-based methods reviewed above using Polyak's (or Polyak-like) step-sizes with the intention of achieving the geometric convergence either require the optimal dual value, whih is unavailable, or explicitly require multipliers to travel infinite distance to guarantee convergence to the optimum λ^*. While the SLR method's advantage is to avoid the need to estimate the optimal dual value, the geometric/linear convergence is only possible outside of a neighborhood of λ^* (Bragin et al. 2015, p. 187). In the following Section 3, a novel Surrogate "Level-Based" Lagrangian Relaxation will be proposed 1. to develop a novel way to determine a "level" estimate of the optimal dual value and 2. to speed up adjustment of the "level" for faster overall convergence.

3. Novel Surrogate "Level-Based" Lagrangian Relaxation

In this section, a novel Surrogate "Level-Based" Lagrangian Relaxation (SLBLR) method is developed with adaptive adjustment of the "level" estimate of the optimal dual value within the Polyak's step-sizing formula for faster convergence. The key is faster (as compared to, for example, Goffin (1977)) multiplier "oscillation detection" based on a novel auxiliary "multiplier-convergence-feasibility" linear constraint satisfaction problem.

"Multiplier-Convergence-Feasibility" Problem to Obtain the Estimate of $q(\lambda^*)$. The premise behind the "Multiplier-Convergence-Feasibility" is the rendition of the result due (Zhao et al. 1999, Theorem 4.1, p. 706):

THEOREM 1. *Under the stepsizing formula*

$$s^k < \gamma \cdot \frac{q(\lambda^*) - L(\tilde{x}^k, \tilde{y}^k, \lambda^k)}{\|g(\tilde{x}^k, \tilde{y}^k)\|^2}, \gamma < 1, \tag{16}$$

such that

$$L(\tilde{x}^k, \tilde{y}^k, \lambda^k) \leq L(\tilde{x}^{k-1}, \tilde{y}^{k-1}, \lambda^k), \tag{17}$$

the multipliers move closer to optimal multipliers λ^* iteration by iteration:

$$\|\lambda^* - \lambda^{k+1}\| < \|\lambda^* - \lambda^k\|. \tag{18}$$

COROLLARY 1. *If*

$$\|\lambda^* - \lambda^{k+1}\| \geq \|\lambda^* - \lambda^k\|, \tag{19}$$

then

$$s^k \geq \gamma \cdot \frac{q(\lambda^*) - L(\tilde{x}^k, \tilde{y}^k, \lambda^k)}{\|g(\tilde{x}^k, \tilde{y}^k)\|^2}. \tag{20}$$

Proof of Corollary 1 Define two predicates

$$A = \left\{ s^k < \gamma \cdot \frac{q(\lambda^*) - L(\tilde{x}^k, \tilde{y}^k, \lambda^k)}{\|g(\tilde{x}^k, \tilde{y}^k)\|^2} \right\}. \tag{21}$$

and

$$B = \left\{ \|\lambda^* - \lambda^{k+1}\| < \|\lambda^* - \lambda^k\| \right\}. \tag{22}$$

From Theorem 1 the following assertion is true $A \Rightarrow B$ and from the Corollary 1 the following assertion is true $\neg B \Rightarrow \neg A$. It remains to prove that both assertions are equivalent. Taking negation of $A \Rightarrow B$ leads to $\neg A \vee B$, and taking negation of $\neg B \Rightarrow \neg A$ leads to $\neg(\neg B) \vee \neg A$, which simplifies to $B \vee \neg A$.

Corollary 2. *If the following auxiliary "multiplier-convergence-feasibility" feasibility problem (with λ being a continuous decision variable: $\lambda \in \mathbb{R}^m$)*

$$\begin{cases} \|\lambda - \lambda^{k_j+1}\| \leq \|\lambda - \lambda^{k_j}\|, \\ \|\lambda - \lambda^{k_j+2}\| \leq \|\lambda - \lambda^{k_j+1}\|, \\ \ldots \\ \|\lambda - \lambda^{k_j+n_j}\| \leq \|\lambda - \lambda^{k_j+n_j-1}\|, \end{cases} \tag{23}$$

admits no feasible solution with respect to λ for some k_j and n_j, then $\exists\, i \in [k_j, k_j + n_j]$ such that

$$s^i \geq \gamma \cdot \frac{q(\lambda^*) - L(\tilde{x}^i, \tilde{y}^i, \lambda^i)}{\|g(\tilde{x}^i, \tilde{y}^i)\|^2}. \tag{24}$$

Proof of Corollary 2 Assume the contrary: $\forall i \in [k_j, k_j + n_j]$ the following holds:

$$s^i < \gamma \cdot \frac{q(\lambda^*) - L(\tilde{x}^i, \tilde{y}^i, \lambda^i)}{\|g(\tilde{x}^i, \tilde{y}^i)\|^2}. \tag{25}$$

By Theorem 1, multipliers approach λ^*, therefore, the "multiplier-convergence-feasibility" problem admits at least one feasible solution - λ^*. Contradiction.

From (25) it follows that $\exists\, \overline{q}_{i,j}$ such that $\overline{q}_{i,j} > q(\lambda^*)$ and the following condition holds:

$$s^i = \gamma \cdot \frac{\overline{q}_{i,j} - L(\tilde{x}^i, \tilde{y}^i, \lambda^i)}{\|g(\tilde{x}^i, \tilde{y}^i)\|^2}. \tag{26}$$

The equation (26) can equivalently be rewritten as:

$$\overline{q}_{i,j} = \gamma \cdot s^i \cdot \|g(\tilde{x}^i, \tilde{y}^i)\|^2 + L(\tilde{x}^i, \tilde{y}^i, \lambda^i). \tag{27}$$

Therefore,

$$\overline{q}_j = \max_{i \in [k_j, k_j+n_j]} \overline{q}_{i,j} > q(\lambda^*). \tag{28}$$

A brief yet important discussion is in order here. The overestimate $\overline{q}_j$ of the dual value $q(\lambda^*)$ is the sought-for "level" value. Unlike previous methods, which require hyper-parameters to obtain "level" values, up to this point, the new method is hyper-parameter-free. Specifically, neither "multipliers-convergence-feasibility" problem (23), nor the computations within (26)-(28) requires hyper-parameter adjustment; following Nedić and Bertsekas (2001b), parameter γ will be chosen as $\gamma = \frac{1}{I}$ (unless specified otherwise), which is the inverse of the number of subproblems, and will not require further adjustments.

To speed up convergence, however, a hyper-parameter $\zeta < 1$ will be introduced to reduce step-sizes as follows:

$$s^{k_{j+1}} = \zeta \cdot \gamma \cdot \frac{\overline{q}_j - L(\tilde{x}^{k_{j+1}}, \tilde{y}^{k_{j+1}}, \lambda^{k_{j+1}})}{\|g(\tilde{x}^{k_{j+1}}, \tilde{y}^{k_{j+1}})\|^2}, \zeta < 1. \tag{29}$$

Subsequently, between iterations k_{j+1} and $k_{j+1} + n_{j+1} - 1$ the step-size is updated dynamically per (29) until the next update $j+2$ when the "multiplier-convergence-feaibility" problem is infeasible again, and the process repeats.

Corollary 3. *The following system of linear inequalities*

$$\begin{cases} 2 \cdot (\lambda - \lambda^{k_j}) \cdot g(\tilde{x}^{k_j}, \tilde{y}^{k_j}) \geq s^{k_j} \cdot \|g(\tilde{x}^{k_j}, \tilde{y}^{k_j})\|^2, \\ 2 \cdot (\lambda - \lambda^{k_j+1}) \cdot g(\tilde{x}^{k_j+1}, \tilde{y}^{k_j+1}) \geq s^{k_j+1} \cdot \|g(\tilde{x}^{k_j+1}, \tilde{y}^{k_j+1})\|^2, \\ \ldots \\ 2 \cdot (\lambda - \lambda^{k_j+n_j-1}) \cdot g(\tilde{x}^{k_j+n_j-1}, \tilde{y}^{k_j+n_j-1}) \geq s^{k_j+n_j-1} \cdot \|g(\tilde{x}^{k_j+n_j-1}, \tilde{y}^{k_j+n_j-1})\|^2, \end{cases} \tag{30}$$

is equivalent to (23).

The above Corollary allows to obtain the "level" value by using LP solvers.

On Improvement of Convergence. To speed up the acceleration of the multipliers divergence detection through the "multiplier-convergence-feasibility" problem (23) is modified, albeit heuristically, in the following way:

$$\begin{cases} \|\lambda - \lambda^{k_j+1}\| \leq \sqrt{1 - 2 \cdot \nu \cdot s^{k_j}} \cdot \|\lambda - \lambda^{k_j}\|, \\ \|\lambda - \lambda^{k_j+2}\| \leq \sqrt{1 - 2 \cdot \nu \cdot s^{k_j+1}} \cdot \|\lambda - \lambda^{k_j+1}\|, \\ \ldots \\ \|\lambda - \lambda^{k_j+n_j}\| \leq \sqrt{1 - 2 \cdot \nu \cdot s^{k_j+n_j-1}} \cdot \|\lambda - \lambda^{k_j+n_j-1}\|. \end{cases} \tag{31}$$

Unlike the problem (23), the problem (31) can no longer be simplified to an LP problem. The potential gain in performance is expected to make up for the increases computational effort to solve (31), which will be verified numerically in Section 4.

Heuristic to Obtain Feasible Solutions. The premise behind the feasible-solution search is that the infeasibility of subproblems solutions with respect to the original problem is caused by a relatively few subproblems. To obtain feasible solutions, only a few subproblem solutions need to be adjusted. However, the exact subproblem solution to be adjusted are unknown. To resolve this difficulty, the procedure developed by Bragin et al. (2019) is used. The main idea behind is to introduce binary "indicator" variables: 1, if subproblem solution is changed/adjusted, and 0, otherwise. The "if" statements are converted to linear constraints and embedded within the original problem, which is then solved. These heuristics only rely on separability of the problem and are universal for general MILP problems. Feasible solutions are typically obtained at the end of the iterative process and no novelties with regards to feasible solution search are claimed in this paper.

4. Numerical Testing

In this Section, a series of examples is considered to test different aspects of the method. In Example 1, a small example with known optimal dual solutions (λ^*) is considered to demonstrate convergence of multipliers. In Example 2, a medium-size generalized assignment problem with 10 machines and 400 jobs is considered to demonstrate the increase of the dual values as well as the decrease of the "level" values. In Example 3, large-scale instances of the generalized assignment problem with 20, 40 and 80 machines and 1600 jobs are considered to demonstrate efficiency, scalability and competitiveness of the method with respect to the best results available thus far in the literature. The new method is implemented within CPLEX 12.10 by using a Dell Precision laptop Intel(R) Xeon(R) E-2286M CPU @ 2.40GHz with installed memory (RAM) of 32.0 GB.

4.1. Example 1: Demonstration of Convergence of Multipliers Based on a Small Example with Known Optimal Multipliers.

Consider the following example (due Bragin et al. (2020)):

$$\min_{x_1,x_2,x_3,x_4,x_5,x_6} \{x_1 + 2x_2 + 3x_3 + x_4 + 2x_5 + 3x_6\}, \tag{32}$$

$$s.t.\ x_1 + 3x_2 + 5x_3 + x_4 + 3x_5 + 5x_6 \geq 26,\ \ 2x_1 + 1.5x_2 + 5x_3 + 2x_4 + 0.5x_5 + x_6 \geq 16. \tag{33}$$

As proved in Bragin et al. (2020), the optimal dual solutions are $\lambda_1^* = 0.6$ and $\lambda_2^* = 0$. Within the SLBLR method, inequality constraints are converted to equality constraints after introducing slack variables. Within this example only, γ will be not chosen as $\gamma = \frac{1}{6}$, rather, a range of values will be tested to demonstrate robustness. In Figure 2, the decrease of the corresponding distances from current multipliers to the optimal multipliers ($\|\lambda^k - \lambda^*\|$) is shown. The results are also compared to those of the Surrogate Lagrangian Relaxation method.

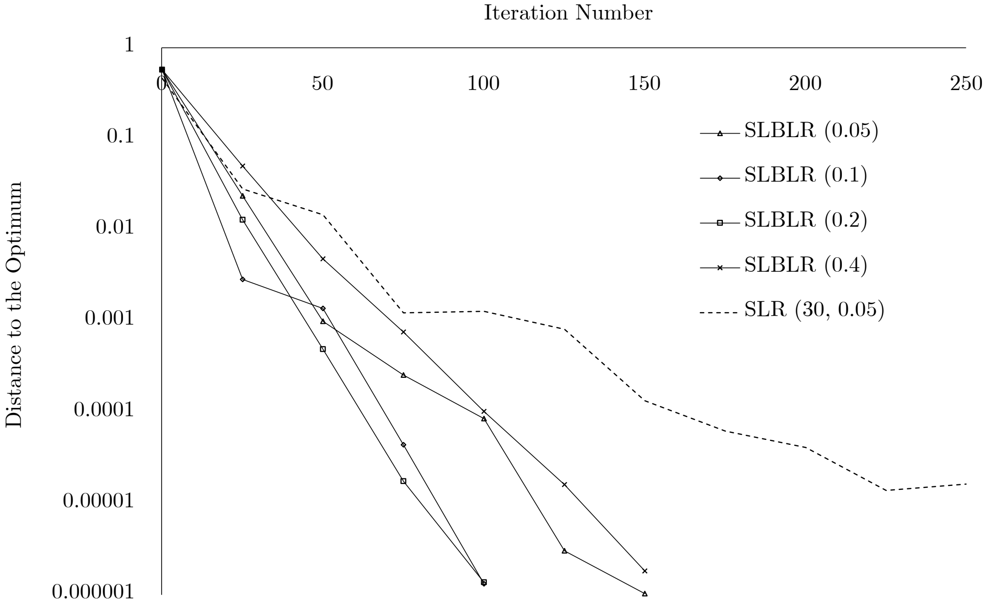

Figure 2 Results for Example 1.

As demonstrated in Figure 2, the SLBLR method converges fast for a wide range of values of γ (from 0.05 to 0.4).

4.2. Example 2: Generalized Assignment Problems.

In this example, generalized assignment problem type D instances from the OR-library Yagiura (2021) will be used. The mathematical formulation of the GAP is as follows:

$$\min_{x_{i,j}} \sum_{i=1}^{I} \sum_{j=1}^{J} g_{i,j} x_{i,j}, \;\; x_{i,j} \in \{0,1\}, \;\; g_{i,j} \geq 0. \tag{34}$$

$$s.t. \;\; \sum_{i=1}^{I} a_{i,j} x_{i,j} \leq b_j, \;\; j=1,\ldots,J, \;\; a_{i,j} \geq 0, \, b_j \geq 0. \tag{35}$$

$$\sum_{j=1}^{J} x_{i,j} = 1, \;\; j=1,\ldots,J,. \tag{36}$$

4.2.1. Convergence for a Medium-Scale Instance. In this subsection, the novel step-sizing method will be tested by using the GAP instance d10400 (with 10 machines and 400 jobs). Performance of the new method will be compared against those by using the following step-sizing formulas: 1. "Contraction mapping" step-sizing (Bragin et al. (2015)); and 2. "Subgradient-level" step-sizing *à la* Goffin and Kiwiel (1999) and Nedić and Bertsekas (2001a). To level the playing field, 1. One subproblem is solved at a time before updating the multipliers in an interleaved-like fashion; and 2. No particular initialization heuristic is used; stepsizes within all the methods are initialized at the value of 0.5 and all the multipliers are initialized at the value of 101. Comparison against the Surrogate Subgradient Method (Zhao et al. (1999)) is not performed because the optimal dual value is unknown. For "contraction mapping" stepsizes, several hyper-parameters were picked from the following ranges: $M \in [30, 50]$ with the step of 5, and $r \in [0.01, 0.1]$ with the step of 0.01 and the best result was obtained for $M = 40$ and $r = 0.05$. For "path-based" stepsizes, parameters β and τ are chosen to be $\frac{1}{2}$, and other hyper-parameters were picked from the following ranges: $R \in [50, 150]$ with a step of 10, and initial $\delta \in [50, 250]$ with a step of 50 and the best result was obtained by using parameters $R = 60$ and $\delta = 100$. Within the new method, the "auxiliary" problem (23) is solved to infer the "level" value and step-sizes (29) are set by using $\zeta = 1$, consequently, no hyper-parameter adjustment was necessary and the results were obtained after one run. In addition to the dual

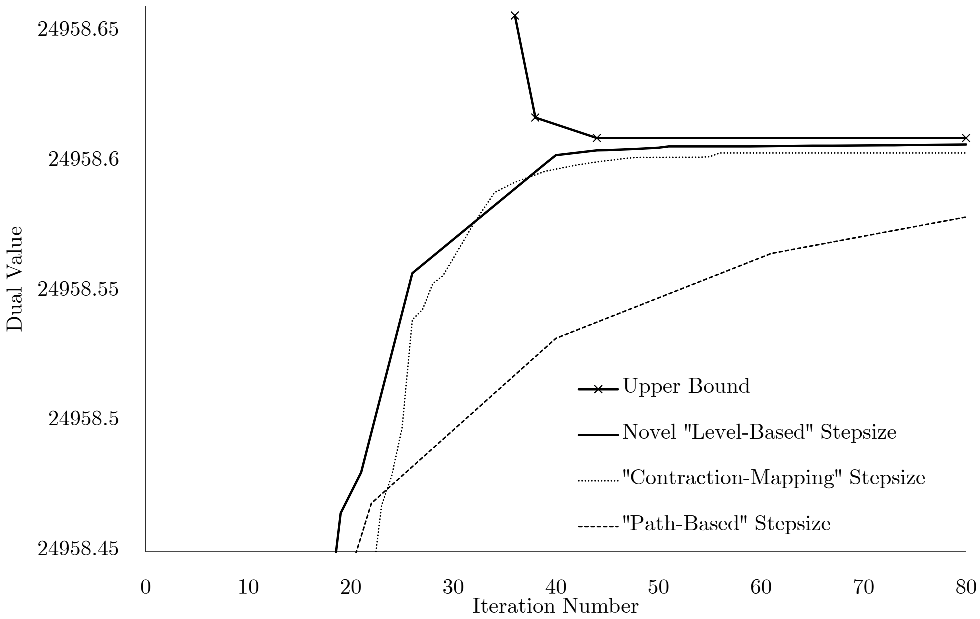

Figure 3 Results for Example 2.

values, the upper bound q_j obtained within the new method is also plotted. The results are shown in Figure 3.

While the total CPU time is not reported since the number of times subproblems are solved is the same, and the total CPU time is roughly the same across all methods, the CPU time involved within the method to solve the "multiplier-convergence-feasibility" problem to obtain the upper bound to calculate stepsizes is worth reporting: 0.187 seconds. This effort more than makes up for with the superior quality of the dual solutions obtained and the avoidance of hyper-parameter adjustment; within (29), parameters γ and δ are chosen to be $\frac{1}{10}$ (the inverse of the number of machines) and 1, respectively.

Herein lies an advantage of the novel SLBLR method: while the choice of hyper-parameters R and δ within the incremental subgradient method as well as the choice of M and r within the Surrogate Lagrangian relaxation is problem-dependent and may require significant effort to determine appropriate values, for this instance of the generalized assignment problem, no hyper-parameter was needed within the new method for achieving a *de facto* superior convergence.

The new method's computational efficiency to obtain feasible solutions as well as to demonstrate robustness with respect to the choice of initial step-sizes and multipliers will be demonstrated next.

4.2.2. Resolution of Large-Scale Instances. Within this example, three largest instances of the generalized assignment problem of type D will be tested: with 20, 40 and 80 machines; all instances contain 1600 jobs. According to Avella et al. (2010), these instances are considered "hard." These instances will be denoted as d201600, d401600, and d801600, respectively.

Robustness of the Method with Respect to Initial Step-sizes. Robustness of the new method with respect to the choice of the initial stepsizes is demonstrated in Table 1. For this trial, multipliers are initialized by using LP dual solutions. Here, and for the rest of the section, the following hyper-parameters are used: within the step-sizing formula (29), parameters γ and δ are chosen to be $\frac{1}{I}$ (the inverse of the number of machines) and $\frac{1}{1.5}$, respectively; within (31), parameter ν is chosen to be 2.

Table 1 Robustness Results for the Instance d201600 with Respect to Initial Step-sizes.

Initial Step-size (s^0)	Feasible Cost	Gap (%)	"Auxiliary" Time (sec)	Total Time (sec)
0.0025	**97825**	0.0037%	123.71	2427.71
0.005	**97825**	0.0037%	6.84	1226.17
0.01	97826	0.0048%	6.96	2143.58
0.02	**97825**	0.0037%	17.10	1195.36
0.04	97826	0.0048%	19.21	1941.55

As demonstrated in Table 1, the method's performance is appreciably stable for the given range of initial step-sizes used.

Robustness of the Method with Respect to Initial Multipliers. Robustness of the new method with respect to the choice of the initial multipliers is demonstrated in Table 2. For this

Table 2 Robustness Results for the Instance d201600 with Respect to Initial Multipliers.

Case Number	Feasible Cost	Total Subproblem Solving Time (sec)	Feasible Solution Search Time (sec)	Multiplier-Convergence-Feasibility Time (sec)	Total Time (sec)
1	97825	1098.74	375.96	22.13	1496.84
2	97826	1009.42	777.16	173.48	1960.07
3	97826	2223.99	221.70	4.54	2450.24
4	97826	2333.55	402.41	4.08	2740.04
5	97826	1002.77	119.91	160.73	1283.42

purpose, the multipliers are initialized randomly by using the uniform distribution $U[90, 110]$. For the testing, the initial step-size 0.02 was used.

Once again, as evidenced from Table 2, the method's performance is fairly stable, exhibiting only a slight degradation of solution accuracy and increase of the CPU time as compared to the case with multipliers initialized by using LP dual solutions.

Scalability of the Method. To test scalability, the three instances d201600, d401600 and d801600 are used. The results are demonstrated in Table 3.

Table 3 Scalability Based on Instances d201600, d401600 and d801600..

Instance Number	Feasible Cost	Gap (%)	"Auxiliary" Time (sec)	Total Time (sec)
d201600	97825	0.0037%	17.10	1195.36
d401600	97105	0.00%	37.57	836.61
d801600	97034	0.00%	141.46	3670.56

As demonstrated in Table 3, the new method is scalable. Moreover, for the two largest instances, the method obtains the optimal solutions.

Competitiveness of the Method. To the best of the author's knowledge, the best results for the three instances have been reported by Sadykov et al. (2015) and the results are summarized in Table 4.

Table 4 Comparison Against the Best Results Currently Available.

Instance	Feasible	Total Time	Feasible	Total Time
	Cost	(hr:min:sec)	Cost Sadykov et al. (2015)	(hr:min:sec) Sadykov et al. (2015)
d201600	97825	0:19:50	97825	0:17:06
d401600	**97105**	0:13:56	97106	0:15:19
d801600	**97034**	1:01:10	97037	3:00:01

As demonstrated in Table 4, for the d201600 instance, the results of the both methods are comparable. For the instance d401600, the new method obtains a better feasible solution and for the instance d801600, the advantage over the existing method is even more pronounced. Even though the comparison in terms of the CPU time is not entirely fair, feasible-cost wise, the new method decisively beats the previous best method. To the best of the author's knowledge, no other reported method obtained optimal results for instances d401600 and d801600.

5. Conclusions

This paper develops the novel MILP solution methodology based on the Lagrangian Relaxation method. Salient features of the new method, inherited from previous version of Lagrangian relaxation, are: 1. reduction of the computational effort required to obtain Lagrangian-multiplier-updating directions and 2. alleviation of zigzagging of multipliers. The key novelty is that new method exploits the underlying geometric-convergence potential inherent to the Polyak's step-sizing formula, tautologically, by using Polyak's formula in conjunction with the novel auxiliary "multiplier-convergence-feasibility" linear programming (LP) constraint satisfaction problem to estimate the "level" value. More importantly, the new method requires neither the optimal dual value, nor the adjustment of hyper-parameters for the estimate of the optimal dual values - the associated the "level" values are determined purely through the simple "auxiliary" optimization; although, one hyper-parameter is needed to govern the reduction of stepsizes per (29), and another is to heuristically speed up convergence per (31). Through testing, it is discovered that the method is robust with respect to the choice of initial stepsizes and multipliers, and the method is computationally efficient and competitive. While "separable" MILP problems are considered, no particular problem characteristics have been used to obtain "level" values, and thus the method has the potential to solve a broader class of mixed-integer programming problems.

References

Avella P, Boccia M, Vasilyev I (2010) A Computational Study of Exact Knapsack Separation for the Generalized Assignment Problem. *Computational Optimization and Applications* 45(3): 543–555.

Bragin MA, Luh PB, Yan JH, Yu N and Stern GA (2015) Convergence of the Surrogate Lagrangian Relaxation Method, *Journal of Optimization Theory & Applications* 164(1): 173–201.

Bragin MA, Luh PB, Yan B, Sun X (2019) A Scalable Solution Methodology for Mixed-Integer Linear Programming Problems Arising in Automation. *IEEE Transactions on Automation Science and Engineering* 16(2): 531–541.

Bragin MA, Yan B, Luh PB (2020) Distributed and Asynchronous Coordination of a Mixed-Integer Linear System via Surrogate Lagrangian Relaxation. *IEEE Transactions on Automation Science and Engineering* 18(3): 1191–1205.

Bragin MA, Dvorkin Y (2021) TSO-DSO Operational Planning Coordination through l_1-Proximal Surrogate Lagrangian Relaxation. *IEEE Transactions on Power Systems*, early access, DOI 10.1109/TPWRS.2021.3101220.

Brännlund U (1993) On Relaxation Methods for Nonsmooth Convex Optimization. *Ph.D. thesis*, Department of Mathematics, Royal Institute of Technology, Stockholm.

Chang SC, Chen C H, Fong IK, Luh PB (1990) Hydroelectric Generation Scheduling with an Effective Differential Dynamic Programming Algorithm. *IEEE Transactions on Power Systems* 5(3): 737–743.

Chang X, Dong M (2017) Stochastic Programming for Qualification Management of Parallel Machines in Semiconductor Manufacturing. *Computers & Operations Research* 79: 49–59.

Chen Y, Wang F, Ma Y, Yao Y (2019) A Distributed Framework for Solving and Benchmarking Security Constrained Unit Commitment with Warm Start. *IEEE Transactions on Power Systems* 35(1): 711–720.

Chen Y, Pan F, Holzer J, Rothberg E, Ma Y, Veeramany A (2020) A High Performance Computing Based Market Economics Driven Neighborhood Search and Polishing Algorithm for Security Constrained Unit Commitment. *IEEE Transactions on Power Systems* 36(1): 292–302.

Cheng Y, Zhang J, Zhang J, Zhao H, Yang L, Zhu H (2021) Small Cell Sleeping and Association for Energy Harvesting Aided Cellular IoT with Full Duplex Self-Backhauls: A Game-Theoretic Approach. *IEEE Internet of Things Journal*, early access, DOI: 10.1109/JIOT.2021.3094218.

Cornuejols G, Fisher ML, Nemhauser GL (1977) Exceptional Paper—Location of Bank Accounts to Optimize Float: An Analytic Study of Exact and Approximate Algorithms. *Management science* 23(8): 789–810.

Cui H, Luo X, Wang Y (2021) Scheduling of Steelmaking-Continuous Casting Process with Different Processing Routes using Effective Surrogate Lagrangian Relaxation Approach and Improved Concave–Convex Procedure. *International Journal of Production Research*, DOI: https://doi.org/10.1080/00207543.2021.1924408.

Cui H, Luo X, Wang Y (2020) Scheduling of Steelmaking-Continuous Casting Process using Deflected Surrogate Lagrangian Relaxation Approach and DC Algorithm. *Computers & Industrial Engineering*, 140, 106271.

Czerwinski CS and Luh PB (1994) Scheduling Products with Bills of Materials using an Improved Lagrangian Relaxation Technique. *IEEE Transactions on Robotics and Automation* 10(2): 99–111.

Erlenkotter D (1978) A Dual-Based Procedure for Uncapacitated Facility Location. *Operations Research* 26(6): 992–1009.

Fisher ML (1973) Optimal Solution of Scheduling Problems Using Lagrange Multipliers: Part I. *Operations Research*, 21(5): 1114–1127

Fisher ML (1976) A Dual Algorithm for the One-Machine Scheduling Problem, *Mathematical Programming* 11(1): 229–251.

Fisher ML (1981) The Lagrangian Relaxation Method for Solving Integer Programming Problems, *Management Science* 27(1): 1–18.

Fisher ML (1985) An Applications Oriented Guide to Lagrangian Relaxation, *Interfaces* 15(2): 10–21.

García Heredia D (2021) New Models and Methods of Mathematical Optimization in Air Traffic Flow Management. *Doctoral Dissertation*, https://e-archivo.uc3m.es/handle/10016/32404, Accessed on: Aug 24, 2021.

Goffin J-L (1977) On Convergence Rates of Subgradient Optimization Methods. *Mathematical Programming* 13(1): 329–347.

Goffin J-L, and Kiwiel KC (1999) Convergence of a Simple Subgradient Level Method. *Math. Program.* 85: 207–211.

Guan X, Luh PB and Zhang L (1995) Nonlinear Approximation Method in Lagrangian Relaxation-Based Algorithms for Hydrothermal Scheduling. *IEEE Transactions on Power Systems* 10(2): 772–778.

Gunn S, Luh PB, Lu X, Hotaling B (2017) Optimizing Guidance for an Active Shooter Event. *In 2017 IEEE International Conference on Robotics and Automation (ICRA)* 4299–4304.

Hämmerle A, Weichhart G (2017) Variable Neighbourhood Search Solving Sub-problems of a Lagrangian Flexible Scheduling Problem. *In ICORES* 234–241.

Held M, and Karp RM (1970) The Traveling-Salesman Problem and Minimum Spanning Trees. *Operations Research* 18(6): 1138–1162.

Held M, and Karp RM (1971) The Traveling-Salesman Problem and Minimum Spanning Trees: Part II. *Mathematical Programming* 1(1): 6–25.

Hoitomt DJ, Luh PB, Max E, and Pattipati KR (1990) Scheduling Jobs with Simple Precedence Constraints on Parallel Machines. *IEEE Control Systems Magazine*, 10(2): 34–40.

Hoitomt DJ, Luh PB, and Pattipati KR (1993) A Practical Approach to Job Shop Scheduling Problems. *IEEE Transactions on Robotics and Automation* 9(1): 1–13.

Hong IH, Chou CC, Lee PK (2019) Admission Control in Queue-Time Loop Production-Mixed Integer Programming with Lagrangian Relaxation (MIPLAR). *Computers & Industrial Engineering* 129: 417–425.

Huang KL, Yang CL, Kuo CM (2020) Plant Factory Crop Scheduling Considering Volume, Yield Changes and Multi-Period Harvests using Lagrangian Relaxation. *Biosystems Engineering* 200: 328–337.

Kaskavelis CA and Caramanis MC (1998) Efficient Lagrangian Relaxation Algorithms for Industry Size Job-Shop Scheduling Problems. *IIE Transactions*, 30(11): 1085–1097.

Kim K, Botterud A, Qiu F (2018) Temporal Decomposition for Improved Unit Commitment in Power System Production Cost Modeling. *IEEE Transactions on Power Systems* 33(5): 5276–5287.

Li X, Zhai Q (2019) Multi-Stage Robust Transmission Constrained Unit Commitment: A Decomposition Framework with Implicit Decision Rules. *International Journal of Electrical Power & Energy Systems* 108: 372–381.

Li X, Zhai Q, Guan X (2020) Robust Transmission Constrained Unit Commitment: A Column Merging Method. *IET Generation, Transmission & Distribution* 14(15): 2968–2975.

Lu X, Blanton H, Gifford T, Tucker A, Olderman N (2021) Optimized Guidance for Building Fires Considering Occupants' Route Choices. *Physica A: Statistical Mechanics and its Applications*, 561, 125247.

Muckstadt J A, Koenig S A (1977) An Application of Lagrangian Relaxation to Scheduling in Power-Generation Systems. *Operations Research*, 25(3): 387–403.

Nedić A, Bertsekas DP (2001) Incremental Subgradient Methods for Nondifferentiable Optimization. *SIAM Journal on Optimization* 12(1): 109–138.

Nedić A, Bertsekas D (2001) Convergence Rate of Incremental Subgradient Algorithms. In *Stochastic optimization: Algorithms and Applications* 223–264, Springer, Boston, MA.

Peng H, Zhou S, Weitze S, Li J, Islam S, Geng T, Li A, Zhang W, Song M, Xie M, Liu H, Ding C (2021) Binary Complex Neural Network Acceleration on FPGA. arXiv preprint arXiv:2108.04811.

Polyak BT (1967) A General Method of Solving Extremum Problems. *Sov. Math. Doklady* 8(3): 593—597.

Polyak BT (1969) Minimization of Unsmooth Functionals. *USSR Comput. Math. Math. Phys.* 9(3): 14—29 (in Russian)

Sadykov R, Vanderbeck F, Pessoa A, Uchoa E (2015) Column Generation Based Heuristic for the Generalized Assignment Problem. XLVII Simpósio Brasileiro de Pesquisa Operacional, Porto de Galinhas, Brazil, 3624-3631.

Shepardson F and Marsten RE (1980) A Lagrangian Relaxation Algorithm for the Two Duty Period Scheduling Problem. *Management Science* 26(3): 274–281.

Shor NZ (1968) On the Rate of Convergence of the Generalized Gradient Method. *Cybernetics* 4(3), 79—80.

Wang L, Zhang Y, Song W, Li Q (2021) Stochastic Cooperative Bidding Strategy for Multiple Microgrids with Peer-to-Peer Energy Trading. *IEEE Transactions on Industrial Informatics*, early access, DOI: 10.1109/TII.2021.3094274.

Wang T, Meng Q, Wang S, Qu X (2021) A Two-Stage Stochastic Nonlinear Integer-Programming Model for Slot Allocation of a Liner Container Shipping Service. *Transportation Research Part B: Methodological* 150: 143–160.

Yagiura M (2021) GAP (Generalized Assignment Problem) Instances. Retrieved (March 8, 2022), http://www.al.cm.is.nagoya-u.ac.jp/∼yagiura/gap/

Yan H, Luh PB, Guan X, Rogan PM (1993) Scheduling of Hydrothermal Power Systems. *IEEE Transactions on Power Systems* 8(3): 1358-1365.

Yan B, Bragin MA, Luh PB (2018) Novel Formulation and Resolution of Job-Shop Scheduling Problems. *IEEE Robotics and Automation Letters* 3(4): 3387–3393.

Yan B, Bragin MA, Luh PB (2020) Tightened Formulation and Resolution of Energy-Efficient Job-Shop Scheduling. *In 2020 IEEE 16th International Conference on Automation Science and Engineering* 741–746.

Zhang D, Mittal K, Wilson J, Luh PB, Fan J, Gupta S (2019) Efficiency and Reliability Joint Optimization of Chiller Plants Based on a Hybrid Model. *IEEE Robotics and Automation Letters* 4(4): 3224–3231.

Zhao X, Luh PB, Wang J (1999) Surrogate Gradient Algorithm for Lagrangian Relaxation. *Journal of Optimization Theory and Applications* 100(3): 699–712.