Improving the Lagrangian Relaxation Approach for Large Job-shop Scheduling

Lei Fang, Peter B. Luh, Haoxun Chen
Department of Electrical & Systems Engineering
University of Connecticut
Storrs, CT 06269-2157
fang@engr.uconn.edu, luh@engr.uconn.edu, hxchen@engr.uconn.edu

Abstract. Lagrangian Relaxation (LR) has recently emerged as a practical approach for job shop scheduling problems of realistic size. The efficiency of the approach, however, depends on how fast the dual problem is solved and how good the feasible solutions are constructed. The purpose of this paper is to address above two issues. First, a "variable target value" method is used to regulate the step size for surrogate subgradient optimization. The target values are updated iteratively whenever necessary, depending on the information obtained in the process of the algorithm. The convergence of the algorithm is proved using practically desirable step size rule. Then based on the insights of the old list scheduling algorithm, the SPT/CR priority index is selected in place of the incremental cost index. Testing results show that these modifications make the LR approach computationally more efficient and five to eight percent of duality gap improvement was obtained for large problems.

1. Introduction

Job shop scheduling problems have been studied over the past several decades [1]. The search for optimal solutions to problems in real manufacturing setting is not practical because of their size and complexity. Thus a fast method that can generate near-optimal solutions is more desirable. Recently, Lagrangian Relaxation-based methods have been developed to solve scheduling problems of realistic size ([2], [3], [4], [5]). The approach begins with a novel problem formulation that is separable; that is, it objective and the complicating constraints are additive. After complicating constraints are relaxed by using Lagrangian multipliers, the original problem can be decomposed into easily solved subproblems. A dual problem derived from the relaxed problem is then formed where the multipliers are iteratively optimized. At the termination of such updating iterations, a primal heuristic is often applied to adjust subproblem solutions to obtain a feasible schedule satisfying all constraints. The efficiency of the approach depends on how fast the dual problem is solved and how good the feasible solutions are constructed.

Since the dual function is polyhedral concave and nondifferentiable, techniques such as the subgradient (SG), bundle, and cutting plane methods are often used [6]. However, the SG ([7], [8]) becomes inefficient for large problems with thousands of subproblems, since it requires minimizing all the subproblems to obtain a search direction. For problems of large sizes, the *surrogate subgradient method* (SSG) was recently developed, where only approximate optimization of the Lagrangian dual is needed to obtain a proper "surrogate subgradient direction" [9].

As far as related step size rules are concerned, they play an important role in governing the practical rate of convergence to optimality. An effective dynamic step size rule suggested in [7] was used by SSG. One assumption here is that the optimal function value is known in advance and it is used in determining the step size of each iteration. In the application of above rule to job shop scheduling, the optimal dual value was estimated by best primal feasible solution obtained in the course of the algorithm. Moreover, the SSG search process may be terminated before the optimal primal solution is achieved because only a limited amount of computation time is allowed in practice. To address this, the optimal dual value can be estimated by a *target value* ([10], [11]). In this paper, the "Variable Target Value Method" (VTVM) presented in [11] will be extended to the context of SSG. The convergence of the algorithm is established by utilizing the special properties of SSG method. The key idea here is that the VTVM checks in every iteration whether the current upper bound of optimal dual can be reduced rather than waiting for many iterations to get a better primal solution to improve the upper bound. The SSG with a variable target value updating procedure will used to solve job shop scheduling problems and improved duality gap can be obtained.

As the dual solution may not be feasible, simple heuristics can be used to adjust relaxed solution to a feasible one. Feasible solutions are also useful in calculating the step sizes in SSG optimization and thus affecting the algorithm convergence. Therefore, an effective heuristic method is highly desirable. A greedy heuristic based on the list scheduling concept were developed in [2]. An incremental change in the objective function was used as tiebreaker to dispatch those operations with the same subproblem solutions. It can be shown that the maximum incremental cost is the minimum slack priority index. Minimum slack rule is not most effective compared to some processing time based rules

[12]. Amongst them, the dispatching rule *SPT/CR* (Shortest Processing Time and Critical Ratio) has been proved quite effective in both static and dynamic job shop setttings ([12], [13]). The merits of SPT/CR include its simplicity and robustness under widely varying conditions in terms of due date tightness and shop load level. Therefore, the SPT/CR priority index is adopted in place of incremental cost priority index in the new heuristics.

The new LR approach taking above modifications is tested on some randomly generated instances as well as on real-life applications. The testing results have been satisfactory, especially for large problems, making our approach practical for shop-floor use.

2. Problem Formulation

Based on our previous work ([2], [4], [5]), the job shop scheduling studied in this paper is formulated as an integer optimization problem. The objective function to be minimized is a weighted sum of part tardiness and earliness penalties (late delivery and inventory), and is part-wise additive, *i.e.*,

$$J \equiv \sum_{i=0}^{I-1} \omega_i T_i^{\,2} + \sum_{i=0}^{I-1} \beta_i E_i^{\,2}. \qquad (2.1)$$

In the above, coefficients ω_i and β_i are tardiness and earlinees weights for part i ($0 \le i \le I-1$). The tardiness T_i is the amount of overdue time, *i.e.*, max ($c_i - d_i$, 0) with c_i the part completion time (the completion time of the part's last operation) and d_i its due date. Earliness E_i is defined as the amount that part beginning time (the beginning time of the part's first operation) leads the desired release time following [2].

The constraints include operation precedence constraints and machine capacity constraints. The operation precedence constraints require that an operation cannot be processed before all its preceding operations have been completed, and are constraints residing within individual parts. Machine capacity constraints require that the number of active operations on a machine type must be less than or equal to the number of machines available at any time, and are complicating constraints that couple parts together. They can in fact be represented in an additive form through a novel formulation, i.e.,

$$\sum_{i=0}^{I-1} \sum_{j=0}^{J_i-1} \delta_{ijkh} \le M_{kh},\ k = 0, \ldots, K\text{-}1;\ h \in H, \qquad (2.2)$$

where δ_{ijkh} is a 0-1 integer variable equal to 1 if operation (i,j), $0 \le j \le J_i - 1$, processed on machine type h at time k, and 0 otherwise.

Since the objective function and the coupling machine capacity constraints are additive, the problem is separable. Our approach is based on LR to exploit this separability of problem formulation.

3. Solution Methodology

Machine capacity constraints are first relaxed by using Lagrangian multipliers. The "relaxed problem" can then be decomposed into smaller and easier part subproblems. These subproblems are solved by using dynamic programming (DP) with precedence constraints embedded in state transitions. Finally, a heuristics is developed to generate feasible schedules satisfying all the constraints based on subproblem solutions.

A. Relaxing Machine Capacity Constraints: By using Lagrangian multipliers λ_{kh} to relax machine capacity constraints, the following relaxed problem is obtained:

$$\min L,\ \text{with } L \equiv \sum_i \left(\omega_i T_i^2 + \beta_i E_i^2\right) + \sum_i \sum_{jkh} \lambda_{kh} \delta_{ijkh} - \sum_{kh} \lambda_{kh} M_{kh}, \qquad (3.1)$$

subject to operation precedence constraints and processing time requirements. By regrouping relevant terms, the relaxed problem can be decomposed into the following part-level subproblems:

$$\min_{\{b_{ij}, h_{ij}\}} L_i,\ \text{with } L_i \equiv \omega_i T_i^2 + \beta_i E_i^2 + \sum_{j=0}^{J_i-1} \sum_{k=b_{ij}}^{c_{ij}} \lambda_{kh}, \qquad (3.2)$$

subject to operation precedence constraints and processing time requirements.

In our previous work [4], [5], it has been shown that each part subproblem is a multistage optimization problem, which can be efficiently solved by using dynamic programming (DP) with polynomial complexity.

B. Dual Problem: Given the optimal subproblem costs $\{L_i^*\}$, the high level dual problem is obtained as:

$$\max_{\lambda_{kh}} q,\ \text{with } q = \sum_i L_i^* - \sum_{kh} \lambda_{kh} M_{kh}, \qquad (3.3)$$

subject to $\lambda_{kh} \ge 0$, k = 0, …, K-1; $h \in H$. (3.4)

In this study, Lagrangian multipliers λ are iteratively updated by a surrogate subgradient method (SSG) to more efficiently solve large problems with many subproblems [9]. The multipliers are updated according to:

$$\lambda^{k+1} = \lambda^k + s^k \tilde{g}^k, \qquad (3.5)$$

where s^k is the step size in iteration k and $\tilde{g}^k$, the surrogate subgradient direction, is calculated based on an approximate solution of dual problem.

Several ways of choosing the step size have been suggested in [7], [8]. Here, the popular one suggested in [7] is adopted:

$$s^k = \gamma^k (q^* - q^k) \Big/ \left\|\tilde{g}^k\right\|^2 \text{ with } 0 < \varepsilon_1 \le \gamma^k \le \varepsilon_2 < 1, \tag{3.6}$$

where q^* is supposed to be the optimal dual value and q^k is the surrogate dual value obtained by approximate minimization of (3.3) in iteration k.

C. *Surrogate Subgradient with Variable Target Value:* In general, the value q^* is not known and the application of the schema (3.6) has been restricted to problems where a good upper or lower bound of q^* is available. To address this, the optimal value q^* can be replaced by a target value, which is an estimate of q^*. This leads to surrogate subgradient method with dynamic step size

$$s^k = \gamma^k (w^k - q^k) \Big/ \left\|\tilde{g}^k\right\|^2, \tag{3.7}$$

with $0 < \varepsilon_1 \le \gamma^k \le \varepsilon_2 < 1$ and w^k being a target value, an estimate of q^*.

A procedure was suggested in [11] to choose the variable target value (VTV) w^k as a convex combination of an adjustable upper bound and current best objective value:

$$w^k = t^k u^k + (1 - t^k) l^k \text{ with } 0 < \varepsilon_3 \le t^k \le 1 - \varepsilon_4 < 1, \tag{3.8}$$

where $w^k \ge q^k$ and $\{w^k\}$ is a monotonically increasing sequence bounded from above, $\{u^k\}$ is nonincreasing and $\{l^k\}$ is nondecreasing with $l^k \ge q^k$. The procedure generates a sequence of target values that converges to q^* in a self-adjusting manner. The above VTVM has a few merits over the schema (3.6) [11].

Rather than waiting for many iterations to get a better primal solution to improve the upper bound, VTVM checks in each iteration whether the current upper bound can be updated. The key idea here is that, if the target value is bigger than the optimal dual value, the path "traveled" by SSG provides a checking condition in the course of the algorithm. Below, the VTVM is formally stated.

SSG with VTV

Step 0 (Initialization). Choose $u^1 > q^*$, $\lambda^1 \in M$ with $M = \{\lambda \mid \lambda \ge 0, q(\lambda) > -\infty\}$. Set $l^1 = q(\lambda^1)$ and w^1 as in (3.8). Consider any optimal solution λ^*. k = 1.

Step 1 (Inner-loop). Set SUM = 0, and $R \ge \|\lambda^k - \lambda^*\|$.

Step 1.1 (Find surrogate subgradient). Obtain a surrogate subgradient $\tilde{g}^k$ at λ^k. If $\tilde{g}^k = 0$, terminate with λ^k optimal.

Step 1.2 (Check the upper bound updating condition). Set $\hat{\lambda} = \lambda^k + s^k \tilde{g}^k$ with s^k from (3.7), and SUM = SUM + $\gamma^k (2 - \gamma^k)(w^k - q^k) / \|\tilde{g}^k\|^2$. Find a nonnegative lower bound $\hat{R}$ of $\left\|\hat{\lambda} - \lambda^*\right\|$. If SUM > $R^2 - \hat{R}^2$, set $u^{k+1} = w^k$ and go to step 2.

Step 1.3 (Move to the new point). Set $\lambda^{k+1} = \hat{\lambda}$, $l^{k+1} = \max\{l^k, q^k\}$, $u^{k+1} = u^k$. Choose t^{k+1} and update w^{k+1} as in (3.8) such that $w^{k+1} \ge w^k$. k = k+1. Go to Step 1.1.

Step 2 (Start with the updated upper bound). Choose any point for λ^{k+1}. Set $l^{k+1} = \max\{l^k, q^k\}$ and $w^{k+1} = l^k$. k = k+1. Go to Step 1.

The following proposition and two lemmas guarantee the convergence of SSG with VTV. One key property of SSG is utilized to prove Proposition 1 and Lemma 2, i.e., the surrogate dual is always grater than or equal to the dual (Proposition 4.1 in [9]).

Proposition 1: Suppose $\lim_{k\to\infty} w^k = w \le q^*$ and $\tilde{g}^k \ne 0$ for all k, then $\lim_{k\to\infty} q(\lambda^k) = w$ and $\lim_{k\to\infty} \lambda^k \in \Lambda(w)$, where $\Lambda(w) = \{\lambda \ge 0 \mid q(\lambda) \ge w\}$.

Proof. See Appendix.

Lemma 1. Suppose $\lim_{k\to\infty} l^k \le q^*$, $\lim_{k\to\infty} u^k \ge q^*$ and $\{w^k\}$ is monotone decreasing. If $\tilde{g}^k \ne 0$ for all k, then:

(i) $q^* + t(u - q^*) \ge w \ge q^*$.

where $t = \limsup_{k\to\infty} t^k$, $u = \lim_{k\to\infty} u^k$, $w = \lim_{k\to\infty} w^k$.

(ii) Moreover if $u > q^*$, then $w > q^*$.

Proof. The proof is similar to the proof of Lemma 1 presented in [11] and is, therefore, omitted.

Lemma 2. Suppose $\{w^k\}$ is monotonically decreasing and $\tilde{g}^k \ne 0$ for all k. Let $\lambda^* \in \Lambda(q)$.

(i) If

$$\sum_{k=1}^{k^0} \gamma^k (2 - \gamma^k)(w^k - q^k)^2 \Big/ \left\|\tilde{g}^k\right\|^2 > \left\|\lambda^1 - \lambda^*\right\|^2 - \left\|\lambda^{k^0+1} - \lambda^*\right\|^2$$

for some k^0, then $w^{k^0} > q^*$.

(ii) If $w^1 > q^*$, then $\sum_k \gamma^k (2 - \gamma^k)(w^k - q^k)^2 \Big/ \left\|\tilde{g}^k\right\|^2$ diverges.

Proof. See Appendix.

*Remark***:** Note that in the process of the inner loop (Step 1) the sequences $\{l^k\}$, $\{u^k\}$, $\{w^k\}$ preserve the premises of Lemma 1. From Lemma 1(ii) and Lemma 2(ii) the condition in Step 1.2 should be satisfied at a finite step once the sequences enter into the inner-loop with a strict upper bound u^k. At each time of leaving the inner-loop, the current target value w^k can be used as an improved upper bound from Lemma 2(i).

To effectively use VTVM in the context of SSG, a good upper bound of the Euclidean distance to an optimal solution needs to be found. It can be shown that for a strong convex function with a strong convexity constant m, $\sqrt{(u^k - q^k)/m}$ can be practically used as an upper bound of $\|\lambda^k - \lambda^*\|$, where u^k is the upper bound of q^* in VTVM.

In this study, the Lagrangian dual is a piece-wise linear function with many facets. A constant (nit in the sense of a strong convex constant) can also be found when the multiplier vector is near to the optimal solution set. However, such a constant need to be adaptively adjusted in the course of the algorithm and its value also depends on the size of the problem according to our extensive testing experience. Generally, a small constant can be selected for a large problem. In the implementation, it is noted that the value $\| \lambda^k - \lambda^* \|^2 - \| \lambda^j - \lambda^* \|^2$ (see Step 1.2) can be well estimated by $\| \lambda^k - \lambda^j \|^2$ if λ^k and λ^j are far from λ^*. Thus, the upper bound updating condition in Step 1.2 can be modified according to above rule of thumb. Moreover, u^k can also be updated by a better primal solution when in the inner loop of the VTVM algorithm. In this case, a restart strategy will be helpful to the convergence of the algorithm.

In our implementation, the parameter γ^k in the step size formula (3.7) is changed adaptively as the algorithm convergences, with the rate of change determined by testing experience (see also [8]). In particular, if no improvement of the lower bound is obtained in several successive iterations, the parameter values are decreased. Although such modification is used, the theoretical convergence of SSG with VTV can still be ensured by removing this modification and stop reducing γ after a certain number of iterations.

The SSG with variable target value is applied to job shop scheduling, leading to three to five percent duality gap improvement. Computational results have been summarized in Section 5.

4. Obtaining Feasible Solutions

With the updating of multipliers at the high level, the optimal part subproblem solutions will provide valuable information for the construction of the feasible primal solution in heuristics. Since the machine capacity constraints are relaxed, the solutions for part subproblems, when put together, may not be feasible.

In a modified version of the list scheduling heuristics of Luh *et al.* [2] (denoted as H1), operations are first arranged in a list in the ascending order of their subproblem beginning times. They are then assigned to machines according to this list as machines become available. When operations have same beginning times, they are sorted in the list according to the incremental cost f(i, j) as defined in [2]:

$$f(i, j) = w_i[(T_i + 1)^2 - T_i^2], \tag{4.1}$$

and the operation with higher incremental cost will be scheduled first.

When the due dates of products are loose, it will be not sufficient just taking into account product tardiness in (4.1) because the tardiness of many products will be zero. If product lateness L (L = c – d) rather than tardiness was used in the heuristic, performance improvement can be expected with respect to loose due dates. Moreover, maximum incremental cost is actually a minimum slack type priority index, since

$$\max[f(i, j)] = \max\{ w_i[(L_i + 1)^2 - L_i^2] \} \approx \max[2 \cdot wi \cdot Li] = \min[2w_i \cdot S_i], \tag{4.2}$$

where S_i is the slack time of product i. It has been shown in the literature that the minimum slack rule is not effective compared to some processing time based rules ([12], [13]).

The simple SPT/CR rule is a combination of the SPT (Shortest Processing Time) and the CR (Critical Ratio) rule, which selects an operation with maximum SPT/CR index to be dispatched next at each time. SPT/CR proves to be quite effective against many criteria ([12], [13]). Thus, a modified SPT/CR rule will be used to order the operations in case of ties of subproblem beginning times. Here, the SPT/CR priority index of the operation (i, j) is defined as

$$SPT/CR_{ij} = \frac{\omega_i}{p_{ij} \cdot \max\left(1, (d_i - t)/r_{ij}\right)} \tag{5.3}$$

where t is the earliest possible beginning time of the operation (i, j) and r_{ij} the remaining processing time of part i including the processing time of the operation (i, j). The earliest possible beginning time t is determined according to the precedence constraints and machine availability. By combining the advantage of the SPT rule on minimizing the weighted complete time criterion (related to work-in-process inventory) and the advantage of the CR rule on minimizing due date related criterions, the SPT/CR rule is an appropriate scheduling rule for the objective of minimizing (2.1).

The above new heuristics (denoted as H2) can be run many times as the multipliers are updated, and the schedule with the lowest cost is chosen as the final schedule.

5. Numerical Testing Results

The method has been implemented in C++ on a Pentium II-400 personal computer. Two examples are presented below to demonstrate the method and to present insights obtained. In the testing, all the multipliers are initialized to zero, and the subgradient algorithm terminated after a fixed amount of CPU time (Example 1), or after a fixed number of iterations for the larger Example 2. In addition, the “time step reduction technique” is used to handle the long time horizon for Examples 2.

Example 1

Several practical job shop scheduling problems are tested by using the SSG+H1 (H1 denotes old heuristics) and the SSGV/TVM+H2 (H2 denotes new heuristics). For both methods tested, the same heuristics H1 are used to obtain feasible solutions. Twenty-five data sets were randomly generated with operation processing times, operation machine types, and part due dates uniformly distributed within appropriate intervals. For each data set, there are 10 (50 or 100) parts each with tardiness weight equal to 1, ten operations per part, and ten machine types in total. Both algorithms are stopped after the same amount of CPU time, and the average results for 25 cases are shown in Table 1.

Table 1. Comparison of SSG/H1 vs. SSG/VTVM/H2

Primal Dim. MT/P/O	Opt. Method	Dual Cost	Primal Cost	Duality Gap (%)	CPU Time (s)
10/10/10	Old	3215	3659	13.8	60
	New	3283	3542	8.0	60
10/50/10	Old	51843	60967	17.6	180
	New	52424	58291	11.2	180
10/100/10	Old	342626	422115	23.2	600
	New	348155	404716	16.2	600

The notation “MT/P/O” provides the number of machine types (MT), the number of parts (P), and the number of operations (O).

From Table 1, we can see that, compared with the old LR approach, the new one can generate better schedules for all problems and the average duality gaps improvement for 10, 50 or 100 parts are 5.8, 6.4 and 7.0, respectively.

Example 2

This example is to illustrate the effectiveness of new method to solve large job shop scheduling problems. Based on sample data sets from Toshiba’s gas insulated switchgear factory, 10 data sets were randomly generated. For each data set, there are 5,000 products with 50,000 operations to be scheduled on 24 machines each with 60 units of a resource. For each data set, the operation processing requirements (resource-hours), product due dates, and tardiness weights deviate from the values of the previous data set by a random percentage uniformly distributed over [-70%, 70%]. The time unit is one hour, the planning horizon is 600 hours, and all multipliers are initialized at zero. Here, the *simplified DP* (SDP) is introduced following the ideas of SSG to reduce the computational requirements for large problems [14]. A surrogate subgradient direction can be obtained without solving all the subproblems; thus, much effort is saved to get a direction for problems with large sizes.

The average duality gaps obtained in 40 minutes with all multiplier initialized at zero are 22.1% for SSG/H1 and 15.9% for SSG/H2 as shown in Fig. 1. The result implies that the new heuristics (H2) can construct better feasible solution than old heuristics (H1) thus improving algorithm convergences. When VTVM is embedded in SSG/H2, another 2% duality gap improvement can be obtained as shown in Fig. 2. The simple but effective step size rule (3.7) plays an important role in improving the SSG algorithm.

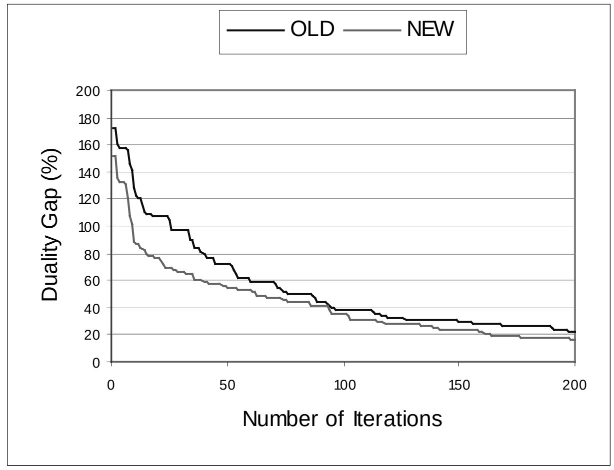

Fig1. Duality gaps obtained by using old heuristics vs. new heuristics

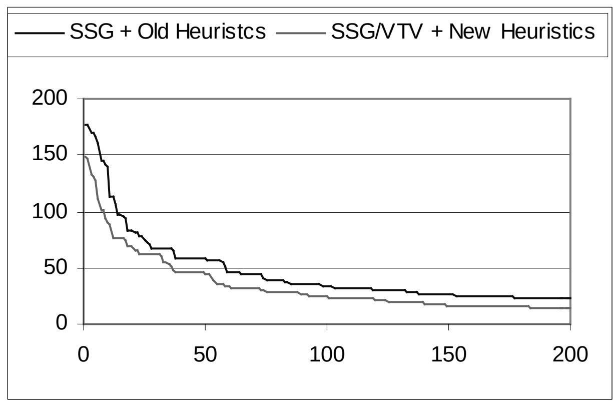

Fig. 2 Duality gaps obtained by using SSG+H1 vs. SSG+H2.

6. Conclusions

In this paper, the surrogate method with variable target value was developed to solve large job shop scheduling problems. The application of SSG with VTVM is, however, not restricted in solving Lagrangian dual and can be extended to any nondifferentiable optimization problems with separable structures. The SSG/VTVM combined with the new heuristics can introduce five to eight percent of duality gap improvements, showing that our new LR approach has the ability to solve practical scheduling problems.

Acknowledgement

This work was supported in part by Toshiba Corporation and the National Science Foundation under Grant DMI-9813176. The authors would like to thank Professor L. S. Thakur and Dr. Yuanhui Zhang of Oracle Corp. (he is formerly with University of Connecticut) for insightful discussions in the development and implementation of improvements of the method.

Appendix

Proof of Proposition 1. Choose any $\lambda \in \Lambda(w)$ [i.e., $q(\lambda) \geq w$]. Then for all k

$$\left\|\lambda^{k+1}-\lambda\right\|^2 = \left\|\lambda^k-\lambda\right\|^2 + \left(s^k\right)^2\left\|\tilde{g}^k\right\|^2 - 2s^k(\lambda-\lambda^k)^T\tilde{g}^k$$

From the definition of surrogate dual [9], we have

$$q(\lambda) \leq q^k + (\lambda-\lambda^k)^T\tilde{g}^k .$$

Using this above, we obtain

$$\begin{aligned}\left\|\lambda^{k+1}-\lambda\right\|^2 &\leq \left\|\lambda^k-\lambda\right\|^2 + \left(s^k\right)^2\left\|g^k\right\|^2 - 2s^k\left(q(\lambda)-q^k\right)\\ &\leq \left\|\lambda^k-\lambda\right\|^2 + \left(s^k\right)^2\left\|g^k\right\|^2 - 2s^k\left(w-q^k\right)\\ &\leq \left\|\lambda^k-\lambda\right\|^2 + \left(s^k\right)^2\left\|g^k\right\|^2 - 2s^k\left(w^k-q^k\right)\\ &= \left\|\lambda^k-\lambda\right\|^2 - \gamma^k(2-\gamma^k)(w^k-q^k)^2\Big/\left\|\tilde{g}^k\right\|^2 .\end{aligned}$$

Hence, $\|\lambda^k - \lambda\|^2$ is monotonically decreasing and converges. The above relation also implies $\lim_{k\to\infty}q(\lambda^k) = w$ since $\left\|\tilde{g}^k\right\|$ is bounded by the boundedness of $\{\lambda^k\}$. Also $\{\lambda^k\}$ has a converging subsequence which converges to a point λ^0 in M. From the continuity of q, $q(\lambda^0) = w$ and $\lambda^0 \in \Lambda(w)$. Since the sequence $\|\lambda^k - \lambda\|^2$ is monotonically decreasing, it is clear that $\lim_{k\to\infty}\lambda^k = \lambda^0$.

Proof of Lemma 2. (i) If $w^{k^0} \leq q^*$, then $w^k \leq q^*$ for all $k \leq k^0$ and from the proof of Proposition 1,

$$\sum_{k=1}^{k^0}\gamma^k(2-\gamma^k)(w^k-\tilde{L}^k)^2\Big/\left\|\tilde{g}^k\right\|^2 \leq \left\|\lambda^1-\lambda^*\right\|^2 - \left\|\lambda^{k^0+1}-\lambda^*\right\|^2$$

(ii) Suppose

$\sum_k\gamma^k(2-\gamma^k)(w^k-q^k)^2\Big/\left\|\tilde{g}^k\right\|^2$ converges. Note that

$$\gamma^k(2-\gamma^k)\,(w^k-q^k)^2\Big/\left\|\tilde{g}^k\right\|^2 \geq \varepsilon_1(1+\varepsilon_2)\,(w^k-q^k)^2\Big/\left\|\tilde{g}^k\right\|^2 ,$$

which implies that $\sum_k (w^k-q^k)^2\Big/\left\|\tilde{g}^k\right\|^2$ converges.

Since $(s^k)^2\left\|\tilde{g}^k\right\|^2 < (w^k-q^k)^2\Big/\left\|\tilde{g}^k\right\|^2$, $\sum_k(s^k)^2\left\|\tilde{g}^k\right\|^2$ converges. Consider

$$\begin{aligned}\left\|\lambda^{k+1}-\lambda^*\right\|^2 &\leq \left\|\lambda^k-\lambda^*\right\|^2 + \left(s^k\right)^2\left\|g^k\right\|^2 - 2s^k\left(q(\lambda^*)-q^k\right)\\ &\leq \left\|\lambda^k-\lambda^*\right\|^2 + \left(s^k\right)^2\left\|g^k\right\|^2,\end{aligned}$$

we have

$$\left\|\lambda^{k+1}-\lambda^*\right\|^2 \leq \left\|\lambda^1-\lambda^*\right\|^2 + \sum_{j=1}^{k}(s^j)^2\left\|\tilde{g}^j\right\|^2 \qquad \forall k.$$

This implies $\{\lambda^k\}$ is a bounded sequence and $\left\|\tilde{g}^k\right\|$ is also bounded by a number G >0. It follows that

$$\gamma^k(2-\gamma^k)\,(w^k-q^k)^2\Big/\left\|\tilde{g}^k\right\|^2 \geq \varepsilon_1(1+\varepsilon_2)\,(w^k-q^k)^2\Big/G^2 > 0 ,$$

which leads to a contradiction. Therefore, the summation diverges.

References

[1] J. Blazewicz, W. Domschke and E. Pesch, "The job shop scheduling problem: conventional and new solution techniques," *European J. of Opn. Res.*, Vol. 93, 1996, pp. 1-33.

[2] D. J. Hoitomt, P.B. Luh, and K.R. Pattipati, "A Practical Approach to Job Shop Scheduling Problems," *IEEE Trans. Robotics &Automation*, Vol. 9, No. 1, 1993, pp. 1-13.

[3] G. Dobson and U. S. Karmarkar, "Large Scale Job Scheduling by Lagrangian Decomposition," *Optimization Models and Concepts in Production Management* (Gordon and Breach Publishers, 1995).

[4] H. Chen, C. Chen and J. M. Proth, "An Improvement of the Lagrangian Relaxation Approach for Job-Shop Scheduling: A Dynamic Programming Method," *IEEE Trans. Robotics & Automation*, Vol. 14, No. 5, 1998, pp. 786-795.

[5] J. Wang, P. B. Luh, X.. Zhao, and J. Wang, "An Optimization-based Algorithm for Job Shop Scheduling," *SADHANA*, Vol. 22, Part 2, 1997, pp. 241-256.

[6] D. P. Bertsekas, *Nonlinear Programming*, Second Edition (Athena Scientific, Belmont, 1999).

[7] B. T. Poljak, "Minimization of Nonsmooth Functionals," *USSR Computational Mathematics and Mathematical Physics*, Vol. 9, 1969, pp. 14-29.

[8] N. Z. Shor, *Minimization Methods for Non-Differentiable Functions* (Springer, Berlin, 1985).

[9] X. Zhao, P.B. Luh and J. Wang, "Surrogate Gradient Algorithms for Lagrangian Relaxation," *J. of Optimization Theory and Application*, Vol. 100, No. 3, March 1999, pp. 699-712.

[10] M. S. Bazaraa and H. D. Sherali, "On the choice of step size in subgradient optimization," *European J. Opn. Res.*, Vol. 7, 1981, pp. 380-388.

[11] S. Kim, H. Ahn and S.C. Cho, "Variable Target Value Subgradient Method," *Math. Programming*, Vol. 49, 1991, pp. 359-369.

[12] E. Kutanoglu and I. Sabuncuoglu, "An Analysis of Heuristics in a Dynamic Job Shop with Weighted Tardiness Objectives," *Int. J. of Product Research*, Vol. 37, No. 1, pp. 165-187.

[13] R. Shafaei and P. Brunn, "Workshop Scheduling Using Practical (inaccurate) Data, Part 1," *Int. J. of Product Research*, Vol. 37, No. 17, 1999, pp. 3913-3925.

[14] Y. Zhang, P. B. Luh, K. Narimatsu, T. Moriya, and T. Shimada, "A Macro-level Scheduling Method Using Lagrangian Relaxation," *Proc. of the 1999 IEEE Int. Conf. SMC*, Tokyo, Japan, Vol. 4, pp. 457-462.