

A Short Proof of Convexity of Step-out Step-in Sequencing Games

Coulter Beeson^a, Neil Olver^{a,*}

^a*Department of Mathematics, London School of Economics and Political Science, United Kingdom*

Abstract

The Step out–Step in sequencing game is a particular example of a game from the sequencing game framework of Curiel, Perderzoli, and Tijs, where coalitions of players in a queue may reorder themselves to improve their overall cost, under some restrictions. Musegaas, Borm and Quant proved, in two papers, that a simple greedy algorithm correctly computes the valuation of a coalition, and that the game is convex. These proofs entail rather involved case analyses; in this note, we give short proofs of both results.

Keywords: Cooperative Game Theory, Sequencing Games, Convexity

1. Introduction

There are many natural settings where people or tasks form a queue and incur some cost for how long they wait. Curiel, Pederzoli, and Tijs [1] took a cooperative game theory perspective and asked the questions “how best can a coalition of players re-arrange themselves to save costs?” and “how can these cost savings be shared?” The main assumption is that players outside of this coalition should be no worse off. How one interprets the notion of “no worse off” gives rise to many different variants of sequencing games. Musegaas, Borm, and Quant [2] introduced the Step out–Step in (SoSi) sequencing game where coalition members cannot move ahead of non-coalition members. They provide a greedy algorithm for computing the optimal cost savings for a coalition in their original paper [2]. In a follow-up work, they prove that the SoSi game is convex [3]. This has important implications for the cooperative game, e.g., that the core is nonempty and can be efficiently computed when the characteristic function can be computed efficiently [4]. Both of these proofs are rather lengthy, and involve a large number of case distinctions to complete the argument.

In this paper we provide greatly simplified proofs of the convexity of SoSi games and the correctness of the greedy algorithm. Throughout we assume

*Corresponding author

Email address: N.Olver@lse.ac.uk (Neil Olver)

familiarity with the basic concepts of cooperative game theory (see, e.g., [5]). We start by formally introducing SoSi games along with other preliminaries. The proofs follow in Section 2.

1.1. Single machine scheduling and sequencing games

Consider the very classical setting of minimizing the weighted sum of completion times, on just a single machine. We are given a set N of n players, processing times $p \in \mathbb{R}_{\geq 0}^N$, and weights $w \in \mathbb{R}_{\geq 0}^N$. Given an ordering $\sigma : N \rightarrow \{1, 2, \dots, n\}$ of the jobs, job j will complete at time $c_j(\sigma) := \sum_{k \in N: \sigma(k) \leq \sigma(j)} p_j$, and incur a cost of $C_j(\sigma) := w_j c_j(\sigma)$. The goal is to minimize the total cost $\sum_{j \in N} C_j(\sigma)$. The celebrated *Smith's rule* [6] states that a minimum cost solution is obtained by ordering the jobs by decreasing *urgency* $u_j := w_j/p_j$.

In a *sequencing game*, this machine scheduling setting is augmented by an initial ordering σ_0 of the players. Further, a set of *admissible orders* available to a coalition S , denoted by $\mathcal{A}(S, \sigma_0)$, is described in some way. Then the *value* of a coalition S is

$$v(S) := C_S(\sigma_0) - \min_{\sigma^* \in \mathcal{A}(S, \sigma_0)} C_S(\sigma^*),$$

where $C_S(\sigma) := \sum_{i \in S} C_i(\sigma)$ denotes the total cost of players in coalition S under the given ordering. In the SoSi game, an order σ is admissible for a coalition S if for all $i \notin S$ and $j \in N$ with $\sigma_0(i) \leq \sigma_0(j)$, we have $\sigma(i) \leq \sigma(j)$.

Musegaas et al. [2] consider the following greedy algorithm for computing $v(S)$ (in fact, they consider a variation of this which is more complicated to describe, but which is equivalent; see Lemma 2). Start with the order $\sigma' = \sigma_0$, and consider each player $i \in S$ in turn, from latest to earliest according to σ_0 . When considering player i , we update σ' by moving the player to a position *later* in the ordering that yields the greatest cost savings for S (breaking ties by choosing the earliest optimal position); if no such move yields an improvement, then σ' is left unchanged. After all players in S have been considered, the algorithm returns $C_S(\sigma_0) - C_S(\sigma')$ as the value of coalition S .

They prove (in two different works) the following theorems.

Theorem 1. [2] *The greedy algorithm correctly computes $v(S)$.*

Theorem 2. [3] *The step-out step-in sequencing game is convex.*

Their proof of convexity relies on the correctness of the greedy algorithm; ours will also. Our proof of Theorem 2, once Theorem 1 has been obtained, is particularly short and simple.

2. Proofs

Preliminaries. We start by defining some useful notation. For any two players $j, k \in N$ let

$$\delta_{jk}(S) := \mathbb{1}_S(k) p_j w_k - \mathbb{1}_S(j) p_k w_j,$$

where $\mathbb{1}_S$ is the indicator function for the set S . The interpretation of this is the decrease in cost for a coalition S obtained by swapping j and k , given that in the current order j directly precedes k . Note that if $u_i > u_j$, then $\delta_{ij} < 0$, and $\delta_{ij} = -\delta_{ji}$. We also define, for $P \subset N$ and $j \in N \setminus P$,

$$\delta_{jP}(S) := \sum_{k \in P} \delta_{jk}(S);$$

if P is a contiguous sets of elements in some order, with j adjacent to and preceding P , this is the cost decrease obtained by moving j to immediately after P . Similarly $\delta_{Pj}(S) := -\delta_{jP}(S)$ is the cost decrease obtained by moving j from immediately after P to immediately before P . We will sometimes omit the explicit dependence on S (writing, for example, δ_{ij} rather than $\delta_{ij}(S)$) when it can cause no ambiguity.

The subgame restricted to the players $T \subset N$, whose value function we denote by v^T , is the sequencing game arising from just the players T with the same weights and processing times, and initially ordered according to the same relative order as σ_0 . For notational convenience, we define $v^T(S) := v^T(S \cap T)$ for all $S \subseteq N$.

For a given coalition S and an order σ , a *component* refers to a maximal subset of S that is contiguous with respect to σ ; that is, an inclusion-wise maximal set of the form $\{j \in N : \sigma(i) \leq \sigma(j) \leq \sigma(k)\} \subseteq S$ for some $i, k \in S$.

The greedy algorithm.

Lemma 1. *For any distinct $j, k \in S$ with $u_j \leq u_k$ and $\ell \in N$, $w_j^{-1}\delta_{j\ell} \geq w_k^{-1}\delta_{k\ell}$.*

Proof. If $\ell \notin S$, then $w_j^{-1}\delta_{j\ell} = -p_\ell = w_k^{-1}\delta_{k\ell}$. If $\ell \in S$, then

$$w_j^{-1}\delta_{j\ell} = w_\ell \cdot \frac{p_j}{w_j} - p_\ell \geq w_\ell \cdot \frac{p_k}{w_k} - p_\ell = w_k^{-1}\delta_{k\ell}. \quad \square$$

Let us say that two orders σ and σ' are *equivalent* for a coalition S if σ' can be obtained from σ via a sequence of swaps of adjacent players in S with equal urgency. Since $\delta_{jk}(S) = p_j p_k (u_k - u_j) = 0$ for $j, k \in S$ with $u_j = u_k$, it follows that $C_S(\sigma) = C_S(\sigma')$ for equivalent orderings σ and σ' .

Lemma 2. *Let σ'_0 be any order obtained from σ_0 by rearranging only players within the same component of S . Let σ_{alg} and σ'_{alg} be the orders determined by the greedy algorithm starting from σ_0 and σ'_0 respectively. Then*

- (i) σ_{alg} and σ'_{alg} are equivalent for coalition S .
- (ii) Any two players in the same component of S respect Smith's rule in both σ_{alg} and σ'_{alg} (i.e., are correctly ordered by urgency).

Proof. It suffices to show that the claim holds if σ'_0 differs from σ_0 by swapping two adjacent players $j, k \in S$, with $\sigma_0(j) < \sigma_0(k)$, since any valid reordering can be obtained via a sequence of such swaps. By symmetry of σ_0 and σ'_0 we may assume $u_j \geq u_k$. Let $W := \{i \in N : \sigma_0(k) < \sigma_0(i)\} = \{i \in N : \sigma'_0(j) < \sigma'_0(i)\}$,

let σ^W be order returned by greedy algorithm for the sub-problem on W . Let $\emptyset = W_0 \subsetneq W_1 \subsetneq \dots \subsetneq W_r = W$ be all prefixes of W under σ^W (i.e., all distinct sets of the form $\{x \in N : \sigma^W(x) \leq \sigma^W(y)\}$ for some y , along with the empty set).

Let t^* be such that $\delta_{kW_{t^*}}$ is maximized, and s^* be such that $\delta_{jW_{s^*}}$ is maximized (breaking ties by choosing t^* and s^* as small as possible). Consider any $t > t^*$, and let $Q = W_t \setminus W_{t^*}$. By the choice of t^* , $\delta_{kQ} = \delta_{kW_t} - \delta_{kW_{t^*}} \leq 0$. Thus

$$\begin{aligned} \delta_{jW_t} &= \delta_{jW_{t^*}} + \delta_{jQ} \\ &\leq \delta_{jW_{t^*}} + \frac{w_j}{w_k} \delta_{kQ} && \text{(by Lemma 1)} \\ &\leq \delta_{jW_{t^*}}. \end{aligned}$$

So $s^* \leq t^*$.

Since also $\delta_{jk} \leq 0$, it follows that when running the greedy algorithm from σ_0 , j will be placed before k in the final ordering. On the other hand, starting from σ'_0 , j will first be placed immediately to the right of W_{s^*} , and k will be placed to the right of j if $u_j > u_k$, or immediately to the left of j if $u_j = u_k$ (since then $s^* = t^*$ and $\delta_{jk} = 0$). That is, the orders obtained by the greedy algorithm after j and k have been considered starting from σ_0 and σ'_0 are equivalent; call these orders σ and σ' respectively. Note that this also proves the second part of the lemma for j and k adjacent.

It suffices to show that inserting the remaining players into σ and σ' does not violate equivalence. The optimal position to place any player ℓ into any two equivalent orderings is the same. If there is a group P of contiguous players with equal urgency, then ℓ will either placed ahead of P (if $\delta_{\ell P} \leq 0$ or behind P (if $\delta_{\ell P} > 0$), irrespective of the ordering of the players in P .

We now prove the second part of the lemma in generality, where j and k are in the same component but not necessarily adjacent. Assume, relabelling if necessary, that $u_j > u_k$ (if $u_j = u_k$, there is nothing to prove). Then as already observed, $\sigma'_{\text{alg}}(j) < \sigma'_{\text{alg}}(k)$. Since σ_{alg} is equivalent to σ'_{alg} by the first part of the lemma, $\sigma_{\text{alg}}(j) < \sigma_{\text{alg}}(k)$ as well. \square

We are now ready to prove that the greedy algorithm correctly computes $v(S)$.

Proof of Theorem 1. By Lemma 2, it suffices to prove the claim under the assumption that each component of S is ordered in σ_0 according to Smith's rule.

Assume inductively that the claim holds for all restrictions of the instance to a strict subset of players. If $S = N$, then σ_0 is already optimal and the greedy algorithm will not change it, so the claim is clear. So assume $S \neq N$, and let q be the earliest player under σ_0 not in S . Let $P \subset S$ be the players preceding q , and $W := N \setminus (P \cup \{q\})$ be all players after q .

Denote by σ the ordering at the point just before any player in P is considered, and by σ_{alg} the final ordering determined by the greedy algorithm. Also let

$\emptyset = W_0 \subsetneq W_1 \subsetneq \dots \subsetneq W_r = W$ be all the prefixes of W under σ . By Lemma 2, the order of the players in P is maintained in σ_{alg} . This means that the cost decrease (if any) attributable to moving player $i \in P$ is $\max\{0, \gamma_i\}$, where

$$\gamma_i := \delta_{iq} + \max_{0 \leq k \leq r} \delta_{iW_k}.$$

In other words, either we don't move i at all and there is no change, or we obtain a cost decrease by moving i to some location after q . So

$$C_S(\sigma_0) - C_S(\sigma_{\text{alg}}) = C_S(\sigma_0) - C_S(\sigma) + \sum_{i \in P} \max\{0, \gamma_i\}. \quad (1)$$

Consider now, for any fixed $T \subseteq P$, the instance restricted to $W \cup T$. The greedy algorithm will again first order W in exactly the same way as σ and then insert each player in T into their optimal positions. The cost decrease associated with player $i \in T$ is then

$$\max_{0 \leq k \leq r} \delta_{iW_k} = \gamma_i - \delta_{iq}.$$

By induction, the result of the greedy algorithm on this instance is optimal, and so we know that

$$v^{W \cup T}(S) = C_S(\sigma_0) - C_S(\sigma) + \sum_{i \in T} (\gamma_i - \delta_{iq}). \quad (2)$$

Consider now an optimal ordering σ^* . Let T^* be the subset of jobs in P that are later than q under σ^* . Let κ be the cost decrease associated with moving players in T^* past players in P : so $\kappa = \sum_{j \in T^*, k \in P \setminus T^* : \sigma_0(j) < \sigma_0(k)} \delta_{jk}$. By assumption $u_j \geq u_k$ and hence $\delta_{jk} \leq 0$ for all the terms in this sum, so $\kappa \leq 0$. Then we have

$$\begin{aligned} v(S) &= v^{W \cup T^*}(S) + \delta_{T^*q} + \kappa \\ &\leq C_S(\sigma_0) - C_S(\sigma) + \sum_{i \in T^*} \gamma_i && \text{by (2)} \\ &\leq C_S(\sigma_0) - C_S(\sigma_{\text{alg}}) && \text{by (1)}. \end{aligned}$$

Thus the cost decreased obtained by the greedy algorithm is at least $v(S)$, and hence equal to $v(S)$. \square

Convexity. Recall that a game is called *convex* if its characteristic function v is *supermodular*, i.e.,

$$v(S \cup T) + v(S \cap T) \geq v(S) + v(T) \quad \forall S, T \subseteq N.$$

A function v is *modular* when we have equality in the above. We will make use of a lemma due to Lovász about supermodular functions. Recall that a set function f is *monotone* if $f(S) \leq f(T)$ for all $S \subseteq T$.

Lemma 3. [7] *Let f and g be supermodular functions, with $f - g$ monotone. Then $\max\{f, g\}$ is supermodular.*

We remark that in the case $f(\emptyset) = g(\emptyset) = 0$ this lemma is trivial as it can be easily argued that $f(S) \geq g(S)$ for all coalitions S . However, we will apply this to more general supermodular functions.

The following lemma is a simple corollary.

Lemma 4. *Let $f_k(S) = \max_{1 \leq r \leq k} \left\{ \sum_{j=1}^r g_j(S) \right\}$, where each $g_j(S)$ is monotone and supermodular. Then f is supermodular.*

Proof. Let $h_r(S) := \sum_{j=1}^r g_j(S)$ for each r . Being the sum of supermodular functions, h_r is again supermodular. Then $f_k(S) = \max\{f_{k-1}(S), h_k(S)\}$. By Lemma 3 it suffices to show that $h_k - f_{k-1}$ is monotone. Let $S \subseteq T$; then

$$h_k(S) - f_{k-1}(S) = \min_{1 \leq r \leq k} \sum_{j=r}^k g_j(S) \leq \min_{1 \leq r \leq k} \sum_{j=r}^k g_j(T) = h_k(T) - f_{k-1}(T),$$

where the inequality follows by monotonicity of the g_j 's. \square

We are now ready to prove the convexity of the SoSi game.

Proof of Theorem 2. Let σ_i be the ordering of players when player $i \in S$ is being considered during the greedy algorithm, and let $f_i(S)$ be the cost saving obtained during this step of the greedy algorithm by moving player i to its optimal location later in the ordering. By the correctness of the greedy algorithm, we have that $v(S) = \sum_{i \in S} f_i(S)$. It suffices then to show that $f_i(S)$ is supermodular for any fixed $i \in S$. Let us write $x \preceq y$ if $\sigma_i(x) \leq \sigma_i(y)$. Then $f_i(S) = \max_{r \succeq i} \left\{ \sum_{i \prec j \preceq r} \delta_{ij}(S) \right\}$. For any $j \in N$, $\delta_{ij}(S) = \mathbb{1}_S(j)p_i w_j - p_j w_i$ is monotone and modular, and hence supermodular. Thus we can apply Lemma 4, and it follows that $v(S)$ is supermodular. \square

Funding. This work was supported by Dutch Science Foundation (NWO) Vidi grant 016.Vidi.189.087.

References

- [1] I. Curiel, G. Pederzoli, S. Tijs, Sequencing games, *European Journal of Operational Research* 40 (3) (1989) 344–351.
- [2] M. Musegaas, P. E. Borm, M. Quant, Step out–step in sequencing games, *European Journal of Operational Research* 246 (3) (2015) 894–906.
- [3] M. Musegaas, P. E. Borm, M. Quant, On the convexity of step out–step in sequencing games, *Top* 26 (1) (2018) 68–109.
- [4] L. S. Shapley, Cores of convex games, *International journal of game theory* 1 (1) (1971) 11–26.

- [5] B. Peleg, P. Sudhölter, Introduction to the theory of cooperative games, Vol. 34, Springer Science & Business Media, 2007.
- [6] W. E. Smith, Various optimizers for single-stage production, Naval Research Logistics Quarterly 3 (1-2) (1956) 59–66. doi:10.1002/nav.3800030106.
URL <https://doi.org/10.1002/nav.3800030106>
- [7] L. Lovász, Submodular functions and convexity, in: Mathematical Programming The State of the Art, Springer, 1983, pp. 235–257.