Technical Appendix

Anonymous submission

A Formal problem definition of the TLSP

The following provides the formal problem definition of the TLSP, taken directly from (Mischek and Musliu 2021):

In the TLSP, a list of projects is given, such that each project contains several tasks. For each project, the tasks must be partitioned into a set of jobs, with some restrictions on the feasible partitions. Then, those jobs must each be assigned a mode, time slots and resources. The properties and feasible assignments for each job are calculated from the tasks contained within.

A solution of TLSP is a schedule consisting of the following parts:

- A list of jobs, composed of one or multiple similar tasks within the same project.
- For each job, an assigned mode, start and end time slots, the employees scheduled to work on the job, and an assignment to a workbench and equipment.

The quality of a schedule is judged according to an objective function that is the weighted sum of several soft constraints and should be minimized. Among others, these include the number of jobs and the total completion time (start of the first job until end of the last) of each project.

Input parameters

A TLSP instance can be split into three parts: The laboratory *environment*, including a list of resources, a list of *projects* containing the tasks that should be scheduled together with their properties and the current state of the *existing schedule*, which might be partially or completely empty.

Environment In the laboratory, resources of different kinds are available that are required to perform tasks:

- *Employees* $e \in E = \{1, \dots, |E|\}$ who are qualified for different types of tasks.
- A number of *workbenches* $b \in B = \{1, \dots, |B|\}$ with different facilities. (These are comparable to machines in shop scheduling problems.)
- Various auxiliary lab *equipment* groups $G_g = \{1, \dots, |G_g|\}$, where g is the group index. These represent sets of similar devices. The set of all equipment groups is called G^*.

The scheduling period is composed of *time slots* $t \in T = \{0, \dots, |T| - 1\}$. Each time slot represents half a day of work.

Tasks are performed in one of several *modes* labeled $m \in M = \{1, \dots, |M|\}$. The chosen mode influences the following properties of tasks performed under it:

- The *speed factor* v_m, which will be applied to the task's original duration.
- The number of *required employees* e_m.

Projects and Tasks Given is a set P of *projects* labeled $p \in \{1, \dots, |P|\}$. Each project contains *tasks* $pa \in A_p$, with $a \in \{1, \dots, |A_p|\}$. The set of all tasks (over all projects) is $A^* = \bigcup_{p \in P} A_p$.

Each task pa has several properties:

- It has a *release date* α_{pa} and both a *due date* $\bar{\omega}_{pa}$ and a *deadline* ω_{pa}. The difference between the latter is that a due date violation only results in a penalty to the solution quality, while deadlines must be observed.
- $M_{pa} \subseteq M$ is the set of *available modes* for the task.
- The task's *duration* d_pa (in time slots, real-valued). Under any given mode $m \in M_{pa}$, this duration becomes $d_{pam} := d_{pa} * v_m$.
- Most tasks must be performed on a workbench. This is indicated by the boolean parameter $b_pa \in \{0, 1\}$. If required, this workbench must be chosen from the set of *available workbenches* $B_{pa} \subseteq B$.
- Similarly, it requires qualified *employees* chosen from $E_{pa} \subseteq E$. The required number depends on the mode. A further subset $E_{pa}^{Pr} \subseteq E_{pa}$ is the set of preferred employees.
- Of each equipment group $g \in G^*$, the task requires r_{pag} devices, which must be taken from the set of *available devices* $G_{pag} \subseteq G_g$.
- A list of direct *predecessors* $\mathcal{P}_{pa} \subseteq A_p$, which must be completed before the task can start. Note that precedence constraints can only exist between tasks in the same project.

Each project's tasks are partitioned into *families* $F_{pf} \subseteq A_p$, where f is the family's index. For a given task pa, f_{pa} gives the task's family. Only tasks from the same family can be grouped into a single job.

Additionally, each family f is associated with a certain *setup time* s_{pf}, which is added to the duration of each job containing tasks of that family.

Finally, it may be required that certain tasks are performed by the same employee(s)[1]. For this reason, each project p may define *linked tasks*, which must be assigned the same employee(s). Linked tasks are given by the equivalence relation $L_p \subseteq A_p \times A_p$, where two tasks pa and pb are linked if and only if $(pa, pb) \in L_p$.

Initial schedule All problem instances include an initial (or base) schedule, which may be completely or partially empty. This schedule can act both as an initial solution and as a baseline, placing limits on the schedules of employees and tasks, in particular by defining fixed assignments that must not be changed.

Provided is a set of jobs J^0, where each job $j \in J^0$ contains the following assignments:

- The tasks in the job: $\dot{A}_j$
  - A *fixed* subset of these tasks $\dot{A}_j^F \subseteq \dot{A}_j$. All fixed tasks of a job in the base schedule must also appear together in a single job in the solution.
- The mode assigned to the job: $\dot{m}_j$
- The start and completion times of the job: $\dot{t}_j^s$ resp. $\dot{t}_j^c$
- The resources assigned to the job:
  - Workbench: $\dot{b}_j$
  - Employees: $\dot{E}_j$
  - Equipment: $\dot{G}_{gj}$ for equipment group g

Except for the tasks, each individual assignment may or may not be present in any given job. Fixed tasks are assumed to be empty, if not given. In all other cases, missing assignments will be referred to using the value ϵ. Time slots and employees can only be assigned if also a mode assignment is given.

A subset of these jobs are the *started jobs* J^{0S}. A started job $j^s \in J^{0S}$ must fulfill the following conditions:

- It must contain at least one fixed task. It is assumed that the fixed tasks of a started job are currently being worked on.
- Its start time must be 0.
- It must contain resource assignments fulfilling all requirements.

A started job's duration does not include a setup time. In the solution, the job containing the fixed tasks of a started job must also start at time 0. Usually, the resources available to the fixed tasks of a started job are additionally restricted to those assigned to the job, to avoid interruptions of ongoing work in case of a rescheduling.

[1]This is used most notably to ensure that documentation is prepared by those employees who also did the tests.

Jobs and Grouping

For various operational reasons, tasks are not scheduled directly. Instead, they are first grouped into larger units called *jobs*.

A single job can only contain tasks from the same project and family.

Jobs have many of the same properties as tasks, which are computed from the tasks that make up a job. The general principle is that within a job, tasks are not explicitly ordered or scheduled; therefore the job must fulfill all requirements of each associated task during its whole duration[2].

Let $J = \{1, \dots, |J|\}$ be the set of all jobs in a solution and $J_p \subseteq J$ be the set of jobs of a given project p. Then for a job $j \in J$, the set of tasks contained in j is $\dot{A}_j$. j has the following properties:

$$\tilde{p}_j \quad \text{and} \quad \tilde{f}_j$$

are the project and family of j.

$$\tilde{\alpha}_j := \max_{pa \in \dot{A}_j} \alpha_{pa}, \quad \tilde{\bar{\omega}}_j := \min_{pa \in \dot{A}_j} \bar{\omega}_{pa}, \quad \tilde{\omega}_j := \min_{pa \in \dot{A}_j} \omega_{pa}$$

are the release date, due date and deadline of j, respectively.

$$\tilde{M}_j := \bigcap_{pa \in \dot{A}_j} M_{pa}$$

is the set of available modes.

$$\tilde{d}_{jm} := \left\lceil (s_{p_j f_j} + \sum_{pa \in \dot{A}_j} d_{pa}) \, * \, v_m \right\rceil$$

is the (integer) duration of the job under mode m. The additional setup time is added to the total duration of the contained tasks.

$$\tilde{b}_j := \max_{pa \in \dot{A}_j} b_{pa}$$

is the required number of workbenches ($\tilde{b}_j \in \{0, 1\}$).

$$\tilde{B}_j := \bigcap_{pa \in \dot{A}_j} B_{pa}$$

are the available workbenches for j.

$$\tilde{E}_j := \bigcap_{pa \in \dot{A}_j} E_{pa}$$

are the employees qualified for j.

$$\tilde{E}_j^{Pr} := \bigcap_{pa \in \dot{A}_j} E_{pa}^{Pr}$$

are the preferred employees of j.

$$\tilde{r}_{jg} := \max_{pa \in \dot{A}_j} r_{pag}$$

are the required units of equipment group g.

$$\tilde{G}_{jg} := \bigcap_{pa \in \dot{A}_j} G_{pag}$$

[2]while this might seem overly restrictive, tasks of the same family usually have equivalent or very similar requirements in practice

are the available devices for equipment group g.

$\tilde{\mathcal{P}}_j \coloneqq \{k \in J \setminus \{j\} : \exists pa \in \dot{A}_j, pb \in \dot{A}_k \text{ s.t. } pb \in \mathcal{P}_{pa}\}$

is the set of predecessor jobs of j. Finally,

$\tilde{L}_p \coloneqq \{(j,k) \in J \times J : j \neq k \wedge \exists pa \in \dot{A}_j, pb \in \dot{A}_k \text{ s.t. } (pa, pb) \in L_p\}$

defines the linked jobs in project p.

In addition, a solution contains the following assignments for each job:

- $\dot{t}^s_j \in T$ the scheduled start time slot
- $\dot{t}^c_j \in T$ the scheduled completion time
- $\dot{m}_j \in M$ the mode in which the job should be performed
- $\dot{b}_j \in B$ the workbench assigned to the job (ϵ if no workbench is required)
- $\dot{E}_j \subseteq E$ the set of employees assigned to the job
- $\dot{G}_{jg} \subseteq G_g$ the set of assigned devices from equipment group g

Constraints

A solution is evaluated in terms of constraints that it should fulfill. *Hard constraints* must all be satisfied in any feasible schedule, while the number and degree of violations of *soft constraints* in a solution give a measure for its quality.

For the purpose of modeling, we introduce additional notation: The set of *active jobs* at time t is defined as $\mathcal{J}_t \coloneqq \{j \in J : \dot{t}^s_j \leq t \wedge \dot{t}^c_j > t\}$.

Hard Constraints

H1: Job assignment. Each task must be assigned to exactly one job.

$$\forall p \in P, pa \in A_p : \quad \exists! j \in J \text{ s.t. } pa \in \dot{A}_j$$

H2: Job grouping. All tasks contained in a job must be from the same project and family.

$$\forall j \in J, pa \in \dot{A}_j : \quad p = \tilde{p}_j \quad f_{pa} = \tilde{f}_j$$

H3: Fixed tasks. Each group of tasks assigned to a fixed job in the base schedule must also be assigned to a single job in the solution.

$$\forall j^0 \in J^0 : \quad \exists j \in J \text{ s.t. } \dot{A}^F_{j^0} \subseteq \dot{A}_j$$

H4: Job duration. The interval between start and completion of a job must match the job's duration.

$$\forall j \in J : \quad \dot{t}^c_j - \dot{t}^s_j = \tilde{d}_{j\dot{m}_j}$$

H5: Time Window. Each job must lie completely within the time window from the release date to the deadline.

$$\forall j \in J : \quad \dot{t}^s_j \geq \tilde{\alpha}_j \quad \dot{t}^c_j \leq \tilde{\omega}_j$$

H6: Task precedence. A job can start only after all prerequisite jobs have been completed.

$$\forall j \in J, k \in \tilde{\mathcal{P}}_j : \quad \dot{t}^c_k \leq \dot{t}^s_j$$

H7: Started jobs. A job containing fixed tasks of a started job in the base schedule must start at time 0.

$$\forall j \in J, j^s \in J^{0S} : \quad t^F_{j^s} = 1 \wedge \dot{A}^F_{j^s} \subseteq \dot{A}_j \implies \dot{t}^s_j = 0$$

H8: Single assignment. At any one time, each workbench, employee and device can be assigned to at most one job.

$$\forall b \in B, t \in T : \quad |\{j \in \mathcal{J}_t : \dot{b}_j = b\}| \leq 1$$
$$\forall e \in E, t \in T : \quad |\{j \in \mathcal{J}_t : e \in \dot{E}_j\}| \leq 1$$
$$\forall g \in G^*, d \in G_g, t \in T : \quad |\{j \in \mathcal{J}_t : d \in \dot{G}_{jg}\}| \leq 1$$

H9a: Workbench requirements. Each job requiring a workbench must have a workbench assigned.

$$\forall j \in J : \quad \dot{b}_j = \epsilon \iff \tilde{b}_j = 0$$

H9b: Employee requirements. Each job must have enough employees assigned to cover the demand given by the selected mode.

$$\forall j \in J : \quad |\dot{E}_j| = e_{\dot{m}_j}$$

H9c: Equipment requirements. Each job must have enough devices of each equipment group assigned to cover the demand for that group.

$$\forall j \in J, g \in G^* : \quad |\dot{G}_{jg}| = \tilde{r}_{jg}$$

H10a: Workbench suitability. The workbench assigned to a job must be suitable for all tasks contained in it.

$$\forall j \in J : \quad \dot{b}_j = \epsilon \vee \dot{b}_j \in \tilde{B}_j$$

H10b: Employee qualification. All employees assigned to a job must be qualified for all tasks contained in it.

$$\forall j \in J : \quad \dot{E}_j \subseteq \tilde{E}_j$$

H10c: Equipment availability. The devices assigned to a job must be taken from the set of available devices for each group.

$$\forall j \in J, g \in G^* : \quad \dot{G}_{jg} \subseteq \tilde{G}_{jg}$$

H11: Linked jobs. Linked jobs must be assigned exactly the same employees.

$$\forall p \in P, (j,k) \in \tilde{L}_p : \quad \dot{E}_j = \dot{E}_k$$

Soft Constraints The following constraints can be used to evaluate the quality of a feasible solution. They arise from the business requirements of our industrial partner.

Each soft constraint violation induces a penalty on the solution quality, denoted as C^i, where i is the soft constraint violated.

S1: Number of jobs. The number of jobs should be minimized.

$$C^{S1} := |J|$$

S2: Employee project preferences. The employees assigned to a job should be taken from the set of preferred employees.

$$\forall j \in J :$$
$$C_j^{S2} := |\{e \in \dot{E}_j : e \notin \tilde{E}_j^{Pr}\}|$$

S3: Number of employees. The number of employees assigned to each project should be minimized.

$$\forall p \in P :$$
$$C_p^{S3} := |\bigcup_{j \in J_p} \dot{E}_j|$$

S4: Due date. The internal due date for each job should be observed.

$$\forall j \in J :$$
$$C_j^{S4} := \max(\dot{t}_j^c - \tilde{\bar{\omega}}_j, 0)$$

S5: Project completion time. The total completion time (start of the first job to end of the last) of each project should be minimized.

$$\forall p \in P :$$
$$C_p^{S5} := \max_{j \in J_p} \dot{t}_j^c \; - \; \min_{j \in J_p} \dot{t}_j^s$$

Constraint S1 favors fewer, longer jobs over more fragmented solutions. This helps reducing overhead (fewer setup periods necessary, rounding of fractional durations), but even more important, it reduces the complexity of the final schedule, both for the employees performing the actual tasks and any human planners in those cases where manual corrections or additions become necessary.

Constraint S2 allows defining "auxiliary" employees, which should only be used if necessary. Typically, these employees usually have other duties, but also possess the required qualifications to perform (some) tasks in the laboratory.

Constraints S3 and S5 reduce overheads by reducing the need for communication (both internal and external), (re-)familiarization with project-specific test procedures and storage space.

Constraint S4 makes the schedule more robust by encouraging tasks to be completed earlier than absolutely required, so they can still be finished on time in case of delays or other disturbances.

The overall solution quality will be determined as the weighted sum over all soft constraint violations.

B Detailed experimental results of multi-objective TLSP solving

Here we provide detailed experimental results for the five different algorithms we evaluated for multi-objective TLSP solving: PSA, PVLNS, PSA-VI, SA, VLNS. Each algorithm was run 5 times on each of the 33 benchmark instances, with a timeout of 2 hours. SA and VLNS are single-threaded algorithms from (Mischek, Musliu, and Schaerf 2021) and (Danzinger et al. 2020), respectively, which we augmented with the functionality to track non-dominated solutions found during the search. PSA, PVLNS, and PSA-VI are described in this paper. For the purpose of this evaluation, they operate on a population of 20 solutions simultaneously, using at most 6 threads. All experiments were conducted on a benchmark server with 256GB RAM and two Intel Xeon CPU E5-2650 v4 processors with 12 logical cores and a frequency of 2.20GHz.

For the analysis and comparison of results, we used several different metrics, capturing different aspects of the found solution sets. Let X be the union of all solutions found for a particular instance across all runs and algorithms.

- The *number of non-dominated solutions* found by the different algorithms (Table 1). This is important since more solutions also mean more options for DMs to fine-tune their preferences.
- The *number of feasible runs*, i.e. runs where the solutions do not contain any conflicts (Table 2).
- The *hypervolume* (Fonseca, Paquete, and Lopez-Ibanez 2006) spanned by the solution set (Table 3). Hypervolumes are calculated relative to the same reference point $\mathbf{z}$ for each instance across all runs and algorithms. To ensure that it is strictly worse than all solutions, we defined the ith coordinate of $\mathbf{z}$ as $z_i = \max_{\mathbf{x} \in X} f_i(\mathbf{x}) + 1$.
- The M_3^* *maximum spread measure* (Zitzler, Deb, and Thiele 2000) of the solution set as an indicator for solution diversity (Table 4). Let $\mathbf{z}^+$ with $z_i^+ = \max_{\mathbf{x} \in X} f_i(\mathbf{x})$ be the nadir point and $\mathbf{z}^-$ with $z_i^- = \min_{\mathbf{x} \in X} f_i(\mathbf{x})$ be the zenith point across all solutions of an instance. For a given solution set U produced by a solver, we then have:

$$M_3^*(U) = \sqrt{\sum_{i=1}^{k} (\max_{\mathbf{x}, \mathbf{y} \in U} \frac{f_i(\mathbf{x}) - f_i(\mathbf{y})}{z_i^+ - z_i^-})^2} \qquad (1)$$

where $z_i^+ - z_i^-$ is a normalization term to ensure that objective value differences fall within the interval [0,1].

C Individual results of the Software Usability Scale survey

Results of the Software Usability Scale (SUS) (Brooke 1996) survey given to the two DMs after interacting with the TLSP MO-Explorer.

The SUS consists of the following 10 statements (taken from (Sauro 2011)), to be graded from 1 (strongly disagree) to 5 (strongly agree):

1. I think that I would like to use this system frequently.

Instance	PSA		PVLNS		PSA-VI		SA		VLNS	
	Best	**Avg**	**Best**	**Avg**	**Best**	**Avg**	**Best**	**Avg**	**Best**	**Avg**
1	15	13.83	4	2.60	16	14.60	15	13.80	15	13.80
2	25	18.33	9	7.00	23	19.40	12	10.80	12	10.80
3	150	113.67	6	4.20	200	132.80	35	26.40	35	26.40
4	201	164.33	16	12.00	211	179.00	49	38.20	49	38.20
5	534	489.33	23	21.20	637	510.40	54	30.20	54	30.20
6	428	369.17	37	34.00	418	373.00	176	132.60	176	132.60
7	543	487.17	42	31.00	640	533.80	86	50.20	86	50.20
8	655	541.67	37	27.00	529	487.80	163	128.80	163	128.80
9	1039	908.67	38	31.80	1107	932.20	170	97.00	170	97.00
10	1267	946.00	99	75.00	1043	847.40	193	141.80	193	141.80
11	756	654.17	57	42.80	785	642.20	158	95.80	158	95.80
12	756	589.83	23	16.80	698	569.40	183	135.60	183	135.60
13	743	635.00	29	21.60	648	601.80	196	139.20	196	139.20
14	822	734.50	29	17.60	709	623.80	81	58.40	81	58.40
15	1119	767.83	62	54.60	794	508.80	119	45.00	119	45.00
16	728	520.17	77	57.80	797	616.40	58	39.80	58	39.80
17	1166	985.00	120	98.20	1113	1036.80	65	28.40	65	28.40
18	1003	792.83	34	21.00	13	10.00	46	18.40	46	18.40
19	611	289.00	4	2.20	1	1.00	20	11.40	20	11.40
20	138	36.33	9	6.20	3	2.00	43	21.80	43	21.80
21	1116	891.17	42	35.20	837	700.40	303	157.60	303	157.60
22	1038	814.00	26	19.80	826	701.20	184	140.40	184	140.40
23	903	662.17	2	1.60	2	1.40	25	17.20	25	17.20
24	979	776.67	53	34.80	926	716.80	55	26.80	55	26.80
25	70	42.83	4	2.20	1	1.00	20	10.60	20	10.60
26	161	63.33	3	2.20	3	1.40	8	5.20	8	5.20
27	1125	892.00	145	76.60	901	643.80	357	80.40	357	80.40
28	510	204.17	63	37.80	620	481.20	113	53.00	113	53.00
29	8	3.00	3	2.00	1	1.00	117	63.00	117	63.00
30	29	16.33	2	1.40	7	5.20	96	24.80	96	24.80
Lab1	109	56.33	2	1.20	2	1.60	172	76.60	172	76.60
Lab2	352	143.00	9	5.20	5	3.60	198	92.40	198	92.40
Lab3	238	177.50	16	12.40	544	370.60	156	87.00	156	87.00
Average	585.97	448.46	34.09	24.76	456.36	371.87	112.91	63.59	112.91	63.59

Table 1: Number of non-dominated solutions found by the different algorithms. Listed are the best and average results across all 5 runs.

Instance	PSA	PVLNS	PSA-VI	SA	VLNS
1	5.00	5.00	5.00	5.00	5.00
2	5.00	5.00	5.00	5.00	5.00
3	5.00	5.00	5.00	5.00	5.00
4	5.00	5.00	5.00	5.00	5.00
5	5.00	5.00	5.00	5.00	5.00
6	5.00	5.00	5.00	5.00	5.00
7	5.00	5.00	5.00	5.00	5.00
8	5.00	5.00	5.00	5.00	5.00
9	5.00	5.00	5.00	5.00	5.00
10	5.00	5.00	5.00	5.00	5.00
11	5.00	5.00	5.00	5.00	5.00
12	5.00	5.00	5.00	5.00	5.00
13	5.00	5.00	5.00	5.00	5.00
14	5.00	5.00	5.00	5.00	5.00
15	5.00	5.00	5.00	5.00	5.00
16	5.00	5.00	5.00	0.00	5.00
17	5.00	5.00	5.00	5.00	5.00
18	5.00	5.00	5.00	5.00	5.00
19	5.00	4.00	0.00	5.00	1.00
20	0.00	3.00	3.00	2.00	0.00
21	5.00	5.00	5.00	5.00	5.00
22	5.00	5.00	5.00	5.00	5.00
23	5.00	0.00	5.00	5.00	0.00
24	5.00	5.00	5.00	5.00	5.00
25	0.00	0.00	0.00	0.00	0.00
26	0.00	0.00	0.00	4.00	0.00
27	5.00	4.00	5.00	5.00	5.00
28	4.00	5.00	5.00	5.00	5.00
29	0.00	0.00	0.00	0.00	0.00
30	0.00	4.00	5.00	3.00	1.00
Lab1	0.00	0.00	1.00	2.00	0.00
Lab2	4.00	5.00	5.00	4.00	2.00
Lab3	4.00	5.00	5.00	5.00	5.00
Average	4.00	4.09	4.21	4.24	3.76

Table 2: Number of runs (out of 5) producing a feasible solution set for the different algorithms.

Instance	PSA		PVLNS		PSA-VI		SA		VLNS	
	Best	**Avg**	**Best**	**Avg**	**Best**	**Avg**	**Best**	**Avg**	**Best**	**Avg**
1	2.0E+04	2.0E+04	1.7E+04	1.6E+04	2.0E+04	1.9E+04	2.1E+04	1.9E+04	1.0E+04	1.0E+04
2	7.4E+04	7.4E+04	7.0E+04	6.5E+04	7.7E+04	7.6E+04	7.6E+04	7.6E+04	5.4E+04	5.3E+04
3	2.2E+06	2.2E+06	2.7E+06	2.6E+06	2.2E+06	2.2E+06	2.5E+06	2.4E+06	2.3E+06	2.3E+06
4	2.9E+06	2.9E+06	3.1E+06	2.9E+06	3.0E+06	2.9E+06	3.2E+06	3.2E+06	2.6E+06	2.4E+06
5	3.5E+07	3.5E+07	7.2E+07	6.8E+07	4.1E+07	3.9E+07	5.3E+07	5.1E+07	5.9E+07	5.5E+07
6	2.3E+07	2.3E+07	2.8E+07	2.7E+07	2.4E+07	2.4E+07	2.8E+07	2.8E+07	2.5E+07	2.2E+07
7	8.3E+07	8.3E+07	1.9E+08	1.8E+08	8.3E+07	8.1E+07	1.5E+08	1.4E+08	1.5E+08	1.2E+08
8	6.0E+07	6.0E+07	1.6E+08	1.5E+08	7.8E+07	7.4E+07	1.2E+08	1.2E+08	1.4E+08	1.3E+08
9	8.6E+08	8.6E+08	3.1E+09	3.0E+09	1.3E+09	1.3E+09	2.4E+09	2.3E+09	2.9E+09	2.6E+09
10	3.4E+09	3.4E+09	2.9E+10	2.8E+10	6.7E+09	6.1E+09	2.2E+10	2.1E+10	2.2E+10	2.1E+10
11	1.2E+09	1.2E+09	5.5E+09	5.1E+09	1.8E+09	1.6E+09	3.7E+09	3.5E+09	4.7E+09	4.3E+09
12	1.0E+09	1.0E+09	3.5E+09	3.2E+09	1.4E+09	1.4E+09	2.6E+09	2.5E+09	3.0E+09	2.7E+09
13	2.0E+08	2.0E+08	2.5E+08	2.4E+08	2.0E+08	2.0E+08	2.4E+08	2.3E+08	2.3E+08	2.3E+08
14	1.4E+08	1.4E+08	3.3E+08	3.0E+08	1.6E+08	1.5E+08	2.5E+08	2.5E+08	2.9E+08	2.7E+08
15	4.8E+09	4.8E+09	6.5E+10	6.1E+10	3.6E+10	2.2E+10	4.7E+10	4.5E+10	5.4E+10	3.9E+10
16	8.9E+09	8.9E+09	7.0E+10	6.5E+10	1.7E+10	1.6E+10			5.6E+10	4.9E+10
17	6.5E+09	6.5E+09	1.3E+11	1.2E+11	5.0E+10	4.0E+10	8.7E+10	8.3E+10	1.1E+11	1.1E+11
18	7.4E+09	7.4E+09	1.2E+11	8.9E+10	6.0E+09	5.3E+09	7.9E+10	7.4E+10	8.7E+10	8.1E+10
19	5.6E+09	5.6E+09	3.4E+10	2.0E+10			6.2E+11	5.5E+11	4.2E+11	4.2E+11
20			1.5E+09	5.3E+08	6.9E+06	3.3E+06	2.3E+10	2.1E+10		
21	2.5E+09	2.5E+09	9.2E+09	8.6E+09	4.4E+09	4.2E+09	8.0E+09	7.7E+09	8.0E+09	7.2E+09
22	1.3E+09	1.3E+09	5.3E+09	4.9E+09	2.4E+09	2.1E+09	4.0E+09	4.0E+09	4.4E+09	4.1E+09
23	4.3E+10	4.3E+10			2.4E+10	9.4E+09	7.6E+11	6.9E+11		
24	4.5E+10	4.5E+10	5.8E+11	5.2E+11	1.9E+11	1.7E+11	4.7E+11	4.3E+11	4.4E+11	4.1E+11
25										
26							8.6E+05	2.5E+05		
27	2.3E+10	2.3E+10	2.3E+12	1.7E+12	4.2E+11	3.7E+11	1.5E+12	1.3E+12	1.8E+12	1.7E+12
28			1.2E+12	1.2E+12	1.8E+11	1.7E+11	8.8E+11	7.4E+11	1.1E+12	1.1E+12
29										
30			6.9E+09	2.0E+09	3.0E+09	1.6E+09	3.1E+11	2.3E+11	7.8E+09	7.8E+09
Lab1					1.0E+00	1.0E+00	4.6E+09	4.4E+09		
Lab2			6.2E+10	4.5E+10	2.5E+10	2.1E+10	8.6E+10	8.5E+10	8.0E+10	6.9E+10
Lab3			2.6E+10	2.2E+10	1.5E+10	1.3E+10	2.5E+10	2.3E+10	2.5E+10	2.2E+10
Average	6.4E+09	6.4E+09	1.7E+11	1.4E+11	3.4E+10	3.0E+10	1.6E+11	1.4E+11	1.6E+11	1.5E+11

Table 3: Hypervolumes spanned by the solution sets found by the different algorithms. Listed are the best and average results across all 5 runs.

Instance	PSA		PVLNS		PSA-VI		SA		VLNS	
	Best	**Avg**	**Best**	**Avg**	**Best**	**Avg**	**Best**	**Avg**	**Best**	**Avg**
1	1.63	1.55	0.68	0.68	1.94	1.68	1.59	1.53	0.92	0.92
2	1.55	1.39	0.90	0.86	1.71	1.54	1.24	0.97	0.53	0.53
3	1.80	1.63	0.35	0.22	1.70	1.60	1.17	1.00	0.00	0.00
4	1.95	1.85	1.07	0.86	2.22	2.01	1.20	1.13	0.51	0.44
5	2.03	1.95	1.09	0.76	1.94	1.89	1.17	1.03	0.80	0.60
6	2.04	1.94	0.81	0.72	1.99	1.92	1.21	1.12	0.58	0.51
7	2.01	1.94	0.98	0.84	1.99	1.91	0.99	0.84	0.36	0.31
8	1.92	1.83	0.76	0.63	2.06	1.96	1.20	1.16	1.31	0.83
9	2.01	1.85	0.44	0.40	2.06	1.98	0.97	0.80	0.74	0.54
10	1.72	1.62	0.94	0.78	1.93	1.87	0.71	0.60	1.22	0.58
11	1.77	1.67	0.81	0.62	1.95	1.89	0.81	0.74	0.95	0.71
12	1.83	1.67	1.12	0.54	1.96	1.85	0.91	0.84	1.01	0.48
13	2.05	1.96	0.69	0.52	2.08	1.98	0.99	0.95	0.73	0.63
14	2.09	1.97	0.88	0.46	2.00	1.92	0.92	0.80	0.52	0.24
15	1.47	1.34	1.17	1.01	1.77	1.44	0.75	0.64	0.90	0.67
16	1.64	1.56	1.08	0.66	1.76	1.69			1.10	0.55
17	1.57	1.48	0.88	0.76	1.94	1.83	0.50	0.22	0.97	0.82
18	1.47	1.33	1.21	0.87	1.14	0.93	0.75	0.42	1.56	0.87
19	1.54	1.25	0.34	0.15			0.22	0.10	0.00	0.00
20			1.47	0.60	0.27	0.09	0.74	0.40		
21	1.97	1.83	0.57	0.47	1.89	1.82	0.86	0.75	0.57	0.50
22	2.04	1.92	0.58	0.49	1.89	1.86	0.95	0.88	0.57	0.32
23	1.30	1.23			0.31	0.11	0.46	0.28		
24	1.21	1.14	1.19	0.75	1.57	1.29	0.62	0.30	1.04	0.60
25										
26							1.02	0.43		
27	1.17	1.04	1.15	0.95	1.58	1.29	0.69	0.21	1.17	1.01
28	1.30	1.07	0.55	0.36	1.77	1.63	0.62	0.30	0.70	0.42
29										
30			0.19	0.05	0.85	0.71	1.01	0.61	0.38	0.38
Lab1					0.00	0.00	0.95	0.85		
Lab2	1.50	0.93	0.47	0.29	0.39	0.33	0.49	0.40	0.27	0.18
Lab3	1.39	1.09	0.64	0.41	1.62	1.23	0.52	0.41	0.77	0.25
Average	1.70	1.56	0.82	0.60	1.60	1.46	0.87	0.69	0.75	0.52

Table 4: M_3^* maximum spread measure achieved by the solution sets found by the different algorithms. Listed are the best and average results across all 5 runs. For each instance and objective, objective values were normalized to the interval [0,1].

2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

Odd numbered questions award $(x-1)$ points, where x is the answer given, while even-numbered questions award $(5-x)$ points. The sum is then multiplied by 2.5, for a total maximum number of 100 points.

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	**Total**
DM1	7.5	7.5	5	5	5	7.5	5	7.5	7.5	7.5	**65**
DM2	10	10	7.5	7.5	10	10	10	10	10	10	**95**
Avg	8.75	8.75	6.25	6.25	7.5	8.75	7.5	8.75	8.75	8.75	**80**

References

Brooke, J. 1996. SUS: a "quick and dirty" usability scale. *Usability evaluation in industry*, 189(3): 189–194.

Danzinger, P.; Geibinger, T.; Mischek, F.; and Musliu, N. 2020. Solving the Test Laboratory Scheduling Problem with Variable Task Grouping. In *International Conference on Planning and Scheduling (ICAPS) 2020*.

Fonseca, C.; Paquete, L.; and Lopez-Ibanez, M. 2006. An Improved Dimension-Sweep Algorithm for the Hypervolume Indicator. In *2006 IEEE International Conference on Evolutionary Computation*, 1157–1163.

Mischek, F.; and Musliu, N. 2021. A local search framework for industrial test laboratory scheduling. *Annals of Operations Research*, 302: 533–562.

Mischek, F.; Musliu, N.; and Schaerf, A. 2021. Local Search Approaches for the Test Laboratory Scheduling Problem with Variable Task Grouping. *Journal of Scheduling*.

Sauro, J. 2011. Measuring Usability with the System Usability Scale (SUS). https://measuringu.com/sus/ (Accessed 2023-07-28).

Zitzler, E.; Deb, K.; and Thiele, L. 2000. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8(2): 173–195.