SIM\_2025-10-28\_141334

# MAIN PARAMETERS

n\_students\_schools = [[40,8],[80,12]]

compare\_solutions = ["SD\_UPON\_DA", "SD\_UPON\_EE", "SD\_UPON\_EADA"]

    # All options are: ["SD\_UPON\_DA", "SD\_UPON\_EE", "SD\_UPON\_EADA"]

n\_iterations\_simul = 10

# DETAILED PARAMETERS

bool\_ColumnGen = False

bool\_identical\_students = False

n\_match = 1000

time\_lim = 600

n\_sol\_pricing = 20

gap\_solutionpool\_pricing = 0.15

MIPGap = 0.05

seed = 0

ALPHA\_INCREMENT = 0.10

#BETA\_INCREMENT = 0.5

alpha = list(np.arange(0, 1.0, ALPHA\_INCREMENT)) + [1.0]

#beta = list(np.arange(0, 1.0, BETA\_INCREMENT)) + [1.0]

#alpha = [0.3]

beta = [0.2,0.6]

print\_intermediate = False

S\_vector = SimulationCG(compare\_solutions, n\_students\_schools, alpha, beta, n\_iterations\_simul, n\_match, time\_lim, seed, n\_sol\_pricing, gap\_solutionpool\_pricing, MIPGap, bool\_ColumnGen, bool\_identical\_students, print\_intermediate)

# SIMULATION RESULTS EXTENDED: ALSO COMPARISON OF # IMPROVING STUDENTS DA AND EE

# SIM\_2025-10-28\_103856

# MAIN PARAMETERS

n\_students\_schools = [[200,20],[250,20],[300,20],[350,20],[400,20],[450,20],[500,20],[550,20],[600,20],[650,20],[700,20],[750,20],[800,20]]

compare\_solutions = ["SD\_UPON\_DA", "SD\_UPON\_EE", "SD\_UPON\_EADA"]

    # All options are: ["SD\_UPON\_DA", "SD\_UPON\_EE", "SD\_UPON\_EADA"]

n\_iterations\_simul = 10

# DETAILED PARAMETERS

bool\_ColumnGen = False

bool\_identical\_students = False

n\_match = 1000

time\_lim = 600

n\_sol\_pricing = 20

gap\_solutionpool\_pricing = 0.15

MIPGap = 0.05

seed = 0

#ALPHA\_INCREMENT = 0.10

#BETA\_INCREMENT = 0.5

#alpha = list(np.arange(0, 1.0, ALPHA\_INCREMENT)) + [1.0]

#beta = list(np.arange(0, 1.0, BETA\_INCREMENT)) + [1.0]

alpha = [0.3]

beta = [0.2]

print\_intermediate = False

S\_vector = SimulationCG(compare\_solutions, n\_students\_schools, alpha, beta, n\_iterations\_simul, n\_match, time\_lim, seed, n\_sol\_pricing, gap\_solutionpool\_pricing, MIPGap, bool\_ColumnGen, bool\_identical\_students, print\_intermediate)

# SIM\_2025-10-28\_083220

# MAIN PARAMETERS

n\_students\_schools = [[50,20],[100,20],[150,20]]

compare\_solutions = ["SD\_UPON\_DA", "SD\_UPON\_EE", "SD\_UPON\_EADA"]

    # All options are: ["SD\_UPON\_DA", "SD\_UPON\_EE", "SD\_UPON\_EADA"]

n\_iterations\_simul = 10

# DETAILED PARAMETERS

bool\_ColumnGen = False

bool\_identical\_students = False

n\_match = 1000

time\_lim = 600

n\_sol\_pricing = 20

gap\_solutionpool\_pricing = 0.15

MIPGap = 0.05

seed = 0

#ALPHA\_INCREMENT = 0.10

#BETA\_INCREMENT = 0.5

#alpha = list(np.arange(0, 1.0, ALPHA\_INCREMENT)) + [1.0]

#beta = list(np.arange(0, 1.0, BETA\_INCREMENT)) + [1.0]

alpha = [0.3]

beta = [0.2]

print\_intermediate = False

S\_vector = SimulationCG(compare\_solutions, n\_students\_schools, alpha, beta, n\_iterations\_simul, n\_match, time\_lim, seed, n\_sol\_pricing, gap\_solutionpool\_pricing, MIPGap, bool\_ColumnGen, bool\_identical\_students, print\_intermediate)

# SIM\_2025-10-27\_140218

# MAIN PARAMETERS

n\_students\_schools = [[80, 8]]

compare\_solutions = ["SD\_UPON\_DA", "SD\_UPON\_EE", "SD\_UPON\_EADA"]

    # All options are: ["SD\_UPON\_DA", "SD\_UPON\_EE", "SD\_UPON\_EADA"]

n\_iterations\_simul = 10

# DETAILED PARAMETERS

bool\_ColumnGen = False

bool\_identical\_students = False

n\_match = 1000

time\_lim = 600

n\_sol\_pricing = 20

gap\_solutionpool\_pricing = 0.15

MIPGap = 0.05

seed = 0

ALPHA\_INCREMENT = 0.10

#BETA\_INCREMENT = 0.5

alpha = list(np.arange(0, 1.0, ALPHA\_INCREMENT)) + [1.0]

#beta = list(np.arange(0, 1.0, BETA\_INCREMENT)) + [1.0]

#alpha = [0.25]

beta = [0.2,0.6]

print\_intermediate = False

S\_vector = SimulationCG(compare\_solutions, n\_students\_schools, alpha, beta, n\_iterations\_simul, n\_match, time\_lim, seed, n\_sol\_pricing, gap\_solutionpool\_pricing, MIPGap, bool\_ColumnGen, bool\_identical\_students, print\_intermediate)

# SIM\_2025-07-29\_160014

n\_students\_schools = [[200,4],[200,8],[200,12], [200, 16], [200, 20], [200, 24], [200, 28], [200, 32]]

compare\_solutions = ["SD\_UPON\_DA", "SD\_UPON\_EE", "SD\_UPON\_EADA"]

    # All options are: ["SD\_UPON\_DA", "SD\_UPON\_EE", "SD\_UPON\_EADA"]

n\_iterations\_simul = 10

n\_match = 1000

time\_lim = 60

n\_sol\_pricing = 20

gap\_solutionpool\_pricing = 0.15

MIPGap = 0.05

bool\_ColumnGen = False

bool\_identical\_students = False

seed = 0

#ALPHA\_INCREMENT = 0.20

#BETA\_INCREMENT = 0.5

#alpha = list(np.arange(0, 1.0, ALPHA\_INCREMENT)) + [1.0]

#beta = list(np.arange(0, 1.0, BETA\_INCREMENT)) + [1.0]

alpha = [0.25]

beta = [0.5]

S\_vector = SimulationCG(compare\_solutions, n\_students\_schools, alpha, beta, n\_iterations\_simul, n\_match, time\_lim, seed, n\_sol\_pricing, gap\_solutionpool\_pricing, MIPGap, bool\_ColumnGen, bool\_identical\_students, True)

# SIM\_2025-07-29\_131019

#n\_students\_schools = [[100,2],[200,4],[400,8], [800, 16], [1600, 32], [3200, 64]]

#n\_students\_schools = [[200,4],[200,8],[200,12], [200, 16], [200, 20], [200, 24], [200, 28], [200, 32]]

n\_students\_schools = [[100, 10]]

compare\_solutions = ["SD\_UPON\_DA", "SD\_UPON\_EE", "SD\_UPON\_EADA"]

    # All options are: ["SD\_UPON\_DA", "SD\_UPON\_EE", "SD\_UPON\_EADA"]

n\_iterations\_simul = 5

n\_match = 1000

time\_lim = 60

n\_sol\_pricing = 20

gap\_solutionpool\_pricing = 0.15

MIPGap = 0.05

bool\_ColumnGen = True

seed = 0

ALPHA\_INCREMENT = 0.20

#BETA\_INCREMENT = 0.5

#alpha = list(np.arange(0, 1.0, ALPHA\_INCREMENT)) + [1.0]

#beta = list(np.arange(0, 1.0, BETA\_INCREMENT)) + [1.0]

alpha = [0.2]

beta = [0.5]

S\_vector = SimulationCG(compare\_solutions, n\_students\_schools, alpha, beta, n\_iterations\_simul, n\_match, time\_lim, seed, n\_sol\_pricing, gap\_solutionpool\_pricing, MIPGap, bool\_ColumnGen, True)

# SIM\_2025-07-29\_110342

#n\_students\_schools = [[100,2],[200,4],[400,8], [800, 16], [1600, 32], [3200, 64]]

#n\_students\_schools = [[200,4],[200,8],[200,12], [200, 16], [200, 20], [200, 24], [200, 28], [200, 32]]

n\_students\_schools = [[10, 4], [20,5], [40, 8]]

compare\_solutions = ["SD\_UPON\_DA", "SD\_UPON\_EE", "SD\_UPON\_EADA"]

    # All options are: ["SD\_UPON\_DA", "SD\_UPON\_EE", "SD\_UPON\_EADA"]

n\_iterations\_simul = 5

n\_match = 1000

time\_lim = 30

n\_sol\_pricing = 10

gap\_solutionpool\_pricing = 0.1

MIPGap = 0.05

bool\_ColumnGen = False

seed = 0

ALPHA\_INCREMENT = 0.20

#BETA\_INCREMENT = 0.5

alpha = list(np.arange(0, 1.0, ALPHA\_INCREMENT)) + [1.0]

#beta = list(np.arange(0, 1.0, BETA\_INCREMENT)) + [1.0]

#alpha = [0.5]

beta = [0.5]

S\_vector = SimulationCG(compare\_solutions, n\_students\_schools, alpha, beta, n\_iterations\_simul, n\_match, time\_lim, seed, n\_sol\_pricing, gap\_solutionpool\_pricing, MIPGap, bool\_ColumnGen, True)

# SIM\_2025-07-29\_093057

#n\_students\_schools = [[100,2],[200,4],[400,8], [800, 16], [1600, 32], [3200, 64]]

#n\_students\_schools = [[200,4],[200,8],[200,12], [200, 16], [200, 20], [200, 24], [200, 28], [200, 32]]

n\_students\_schools = [[20,4], [40, 8]]

compare\_solutions = ["SD\_UPON\_DA", "SD\_UPON\_EE", "SD\_UPON\_EADA"]

    # All options are: ["SD\_UPON\_DA", "SD\_UPON\_EE", "SD\_UPON\_EADA"]

n\_iterations\_simul = 3

n\_match = 1000

time\_lim = 30

n\_sol\_pricing = 10

gap\_solutionpool\_pricing = 0.1

MIPGap = 0.05

bool\_ColumnGen = False

seed = 0

ALPHA\_INCREMENT = 0.25

#BETA\_INCREMENT = 0.5

alpha = list(np.arange(0, 1.0, ALPHA\_INCREMENT)) + [1.0]

#beta = list(np.arange(0, 1.0, BETA\_INCREMENT)) + [1.0]

#alpha = [0.5]

beta = [0.5]

S\_vector = SimulationCG(compare\_solutions, n\_students\_schools, alpha, beta, n\_iterations\_simul, n\_match, time\_lim, seed, n\_sol\_pricing, gap\_solutionpool\_pricing, MIPGap, bool\_ColumnGen, True)

# SIM\_2025-07-28\_141746.csv

#n\_students\_schools = [[100,2],[200,4],[400,8], [800, 16], [1600, 32], [3200, 64]]

#n\_students\_schools = [[200,4],[200,8],[200,12], [200, 16], [200, 20], [200, 24], [200, 28], [200, 32]]

n\_students\_schools = [[20,4], [40, 8]]

compare\_solutions = ["SD\_UPON\_DA", "SD\_UPON\_EADA"]

    # All options are: ["SD\_UPON\_DA", "SD\_UPON\_EE", "SD\_UPON\_EADA"]

n\_iterations\_simul = 3

n\_match = 1000

time\_lim = 30

n\_sol\_pricing = 10

gap\_solutionpool\_pricing = 0.1

MIPGap = 0.05

bool\_ColumnGen = False

seed = 0

ALPHA\_INCREMENT = 0.5

#BETA\_INCREMENT = 0.5

alpha = list(np.arange(0, 1.0, ALPHA\_INCREMENT)) + [1.0]

#beta = list(np.arange(0, 1.0, BETA\_INCREMENT)) + [1.0]

#alpha = [0.5]

beta = [0.5]

S\_vector = SimulationCG(compare\_solutions, n\_students\_schools, alpha, beta, n\_iterations\_simul, n\_match, time\_lim, seed, n\_sol\_pricing, gap\_solutionpool\_pricing, MIPGap, bool\_ColumnGen, True)

# SIM\_2025-06-30\_141452

#n\_students\_schools = [[100,2],[200,4],[400,8], [800, 16], [1600, 32], [3200, 64]]

n\_students\_schools = [[200,4],[200,8],[200,12], [200, 16], [200, 20], [200, 24], [200, 28], [200, 32]]

#n\_students\_schools = [[200,4],[200,8],[200,12], [200, 16], [200, 20], [200, 24], [200, 28], [200, 32]]

n\_students\_schools = [[20,4], [40, 8]]

compare\_solutions = ["SD\_UPON\_DA", "SD\_UPON\_EADA"]

    # All options are: ["SD\_UPON\_DA", "SD\_UPON\_EE", "SD\_UPON\_EADA"]

n\_iterations\_simul = 3

n\_match = 1000

time\_lim = 30

n\_sol\_pricing = 10

gap\_solutionpool\_pricing = 0.1

MIPGap = 0.05

bool\_ColumnGen = False

seed = 0

ALPHA\_INCREMENT = 0.5

#BETA\_INCREMENT = 0.5

alpha = list(np.arange(0, 1.0, ALPHA\_INCREMENT)) + [1.0]

#beta = list(np.arange(0, 1.0, BETA\_INCREMENT)) + [1.0]

#alpha = [0.5]

beta = [0.5]

S\_vector = SimulationCG(compare\_solutions, n\_students\_schools, alpha, beta, n\_iterations\_simul, n\_match, time\_lim, seed, n\_sol\_pricing, gap\_solutionpool\_pricing, MIPGap, bool\_ColumnGen, True)

# SIM\_2025-06-30\_141452

#n\_students\_schools = [[100,2],[200,4],[400,8], [800, 16], [1600, 32], [3200, 64]]

n\_students\_schools = [[200,4],[200,8],[200,12], [200, 16], [200, 20], [200, 24], [200, 28], [200, 32]]

n\_iterations\_simul = 2

n\_match = 1000

time\_lim = 30

n\_sol\_pricing = 10

gap\_pricing = 0.1

bool\_ColumnGen = False

seed = 0

ALPHA\_INCREMENT = 1

BETA\_INCREMENT = 0.2

alpha = list(np.arange(0, 1.0, ALPHA\_INCREMENT)) + [1.0]

#beta = list(np.arange(0, 1.0, BETA\_INCREMENT)) + [1.0]

alpha = [0.5]

beta = [0.5]

S\_vector = SimulationCG(n\_students\_schools, alpha, beta, n\_iterations\_simul, n\_match, time\_lim, n\_sol\_pricing, gap\_pricing, bool\_ColumnGen, seed, True)

# SIM\_2025-06-30\_112104

n\_students\_schools = [[100,4],[200,8],[400,16], [800, 32]]

n\_iterations\_simul = 2

n\_match = 1000

time\_lim = 30

n\_sol\_pricing = 10

gap\_pricing = 0.1

bool\_ColumnGen = False

seed = 0

ALPHA\_INCREMENT = 1

BETA\_INCREMENT = 0.2

alpha = list(np.arange(0, 1.0, ALPHA\_INCREMENT)) + [1.0]

#beta = list(np.arange(0, 1.0, BETA\_INCREMENT)) + [1.0]

alpha = [0.5]

beta = [0.5]

S\_vector = SimulationCG(n\_students\_schools, alpha, beta, n\_iterations\_simul, n\_match, time\_lim, n\_sol\_pricing, gap\_pricing, bool\_ColumnGen, seed, True)

# SIM\_2025-06-26\_163524

#n\_students\_schools = [[100,2],[200,4],[400,8], [800, 16], [1600, 32], [3200, 64]]

n\_students\_schools = [[100,20],[200,20],[400,20], [800, 20]]

n\_iterations\_simul = 2

n\_match = 1000

time\_lim = 30

n\_sol\_pricing = 10

gap\_pricing = 0.1

bool\_ColumnGen = False

seed = 0

ALPHA\_INCREMENT = 1

BETA\_INCREMENT = 0.2

alpha = list(np.arange(0, 1.0, ALPHA\_INCREMENT)) + [1.0]

#beta = list(np.arange(0, 1.0, BETA\_INCREMENT)) + [1.0]

alpha = [0.5]

beta = [0.5]

S\_vector = SimulationCG(n\_students\_schools, alpha, beta, n\_iterations\_simul, n\_match, time\_lim, n\_sol\_pricing, gap\_pricing, bool\_ColumnGen, seed, True)

# SIM\_2025-06-23\_174124

n\_students = [30]

n\_schools = [6]

n\_iterations\_simul = 5

n\_match = 1000

time\_lim = 30

seed = 0

ALPHA\_INCREMENT = 0.2

BETA\_INCREMENT = 0.2

alpha = list(np.arange(0, 1.0, ALPHA\_INCREMENT)) + [1.0]

#beta = list(np.arange(0, 1.0, BETA\_INCREMENT)) + [1.0]

beta = [0.5]

print(alpha)

S\_vector = SimulationCG(n\_students, n\_schools, alpha, beta, n\_iterations\_simul, n\_match, time\_lim, seed, True)

# SIM\_2025-06-23\_163321

n\_students = [30]

n\_schools = [6]

n\_iterations\_simul = 2

n\_match = 1000

time\_lim = 30

seed = 0

ALPHA\_INCREMENT = 0.2

BETA\_INCREMENT = 0.2

alpha = list(np.arange(0, 1.0, ALPHA\_INCREMENT)) + [1.0]

#beta = list(np.arange(0, 1.0, BETA\_INCREMENT)) + [1.0]

beta = [0.5]

print(alpha)

S\_vector = SimulationCG(n\_students, n\_schools, alpha, beta, n\_iterations\_simul, n\_match, time\_lim, seed, True)

# SIM\_2025-06-23\_113926

n\_students = [500]

n\_schools = [15]

n\_iterations\_simul = 1

n\_match = 1000

time\_lim = 300

seed = 0

ALPHA\_INCREMENT = 1

BETA\_INCREMENT = 0.2

alpha = [0.5]

beta = list(np.arange(0, 1.0, BETA\_INCREMENT)) + [1.0]

print(alpha)

S\_vector = SimulationCG(n\_students, n\_schools, alpha, beta, n\_iterations\_simul, n\_match, time\_lim, seed, True)

# SIM\_2025-06-20\_161207

n\_students = [50]

n\_schools = [6]

n\_iterations\_simul = 10

n\_match = 1000

time\_lim = 30

seed = 0

ALPHA\_INCREMENT = 0.1

BETA\_INCREMENT = 0.2

alpha = list(np.arange(0, 1.0, ALPHA\_INCREMENT)) + [1.0]

beta = list(np.arange(0, 1.0, BETA\_INCREMENT)) + [1.0]

print(alpha)

S\_vector = SimulationCG(n\_students, n\_schools, alpha, beta, n\_iterations\_simul, n\_match, time\_lim, seed, True)

# SIM\_2025-06-20\_133036

n\_students = [30]

n\_schools = [6]

n\_iterations\_simul = 5

n\_match = 1000

time\_lim = 30

seed = 0

ALPHA\_INCREMENT = 0.5

BETA\_INCREMENT = 0.5

alpha = list(np.arange(0, 1.0, ALPHA\_INCREMENT)) + [1.0]

beta = list(np.arange(0, 1.0, BETA\_INCREMENT)) + [1.0]

print(alpha)

S\_vector = SimulationCG(n\_students, n\_schools, alpha, beta, n\_iterations\_simul, n\_match, time\_lim, seed, True)