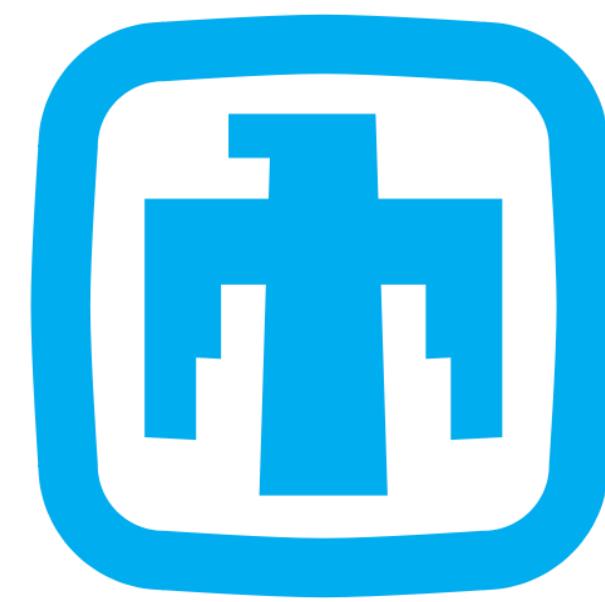




Advances in Generalized Disjunctive and Mixed-Integer Nonlinear Programming Algorithms and Software for Superstructure Optimization

David E. Bernal¹, Yunshan Liu¹, Michael L. Bynum², Carl D. Laird¹, John D. Sirola², and Ignacio E. Grossmann¹

¹Department of Chemical Engineering, Carnegie Mellon University, USA ²Discrete Mathematics and Optimization, Sandia National Laboratories, USA

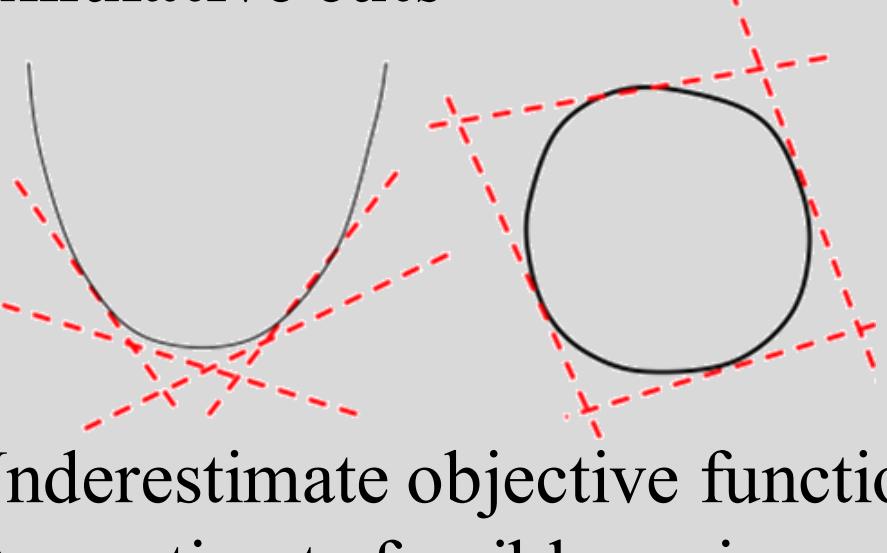
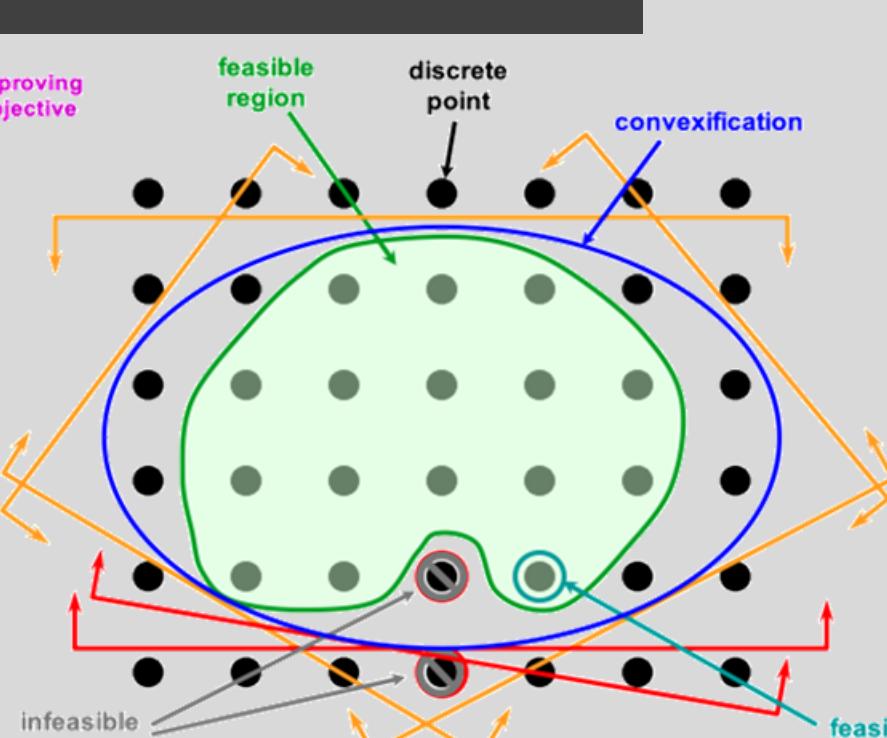


Sandia
National
Laboratories

Motivation

Goal	Present recent advances in optimization algorithms and software to solve process superstructure problems in a way that is easily accessible to users and extensible to advanced modeling systems.
State-of-the-art	<ul style="list-style-type: none"> Process units implemented in programming languages with interface to optimization solvers. Modeling of superstructure done through Mixed-integer nonlinear programming (MINLP).
Challenges	<p>Challenging Optimization problems</p> <ul style="list-style-type: none"> Requirement for versatile modeling framework combining discrete and continuous variables, with nonlinear constraints. <p>Diverse and incompatible solutions tools</p> <ul style="list-style-type: none"> Requirement for specialized solution methods <p>Modeling needs to be accessible to users and allow for advanced optimization algorithmic capabilities.</p>
Approach	Introduce MINLP and Generalized Disjunctive Programming (GDP) open-source modeling tools and solvers in Python to implement process superstructure and release examples in library GDPLib .

Mixed-Integer Nonlinear Programming

Formulation	Overview and solution approach
$\min_{x,y} f(x,y)$ $s.t. g(x,y) \leq 0$ $x \in X \subseteq \mathbb{R}^{n_x}$ $y \in Y \subseteq \mathbb{Z}^{n_y}$	Optimization problems with algebraic nonlinear constraints in terms of discrete and continuous variables . Solution methods rely on solving easier subproblems leading to bounds of optimal solution: <ul style="list-style-type: none"> Relaxations: on larger feasible spaces give optimistic or Lower bound on the optimum Restrictions: evaluating only a subset of the possible solutions provide a feasible or Upper bound on optimum
Convex MINLP	
<p>Guaranteed to converge by cumulative cuts</p>  <p>Underestimate objective function Overestimate feasible region</p>	
Nonconvex MINLP	
 <p>Requires convexification ($g \rightarrow \hat{g}$) to converge</p>	
Branch-and-bound	
<ul style="list-style-type: none"> Start from continuous relaxation Systematically enforce integer constraints Strengthened with inequalities or cuts Really successful for the linear case 	
Decomposition-methods	
<ul style="list-style-type: none"> Iterative solution of restrictions and approximations to find feasible solutions. If approximations are valid relaxations, and subproblems are solved globally, global optimality for MINLP can be guaranteed. Examples: <ul style="list-style-type: none"> Outer-Approximation Extended-Cutting Planes Generalized Benders Decomposition 	

Generalized Disjunctive Programming

Modeling framework	Solution methods
<ul style="list-style-type: none"> Alternative to traditional MINLP modeling Preserves logical structure and ability to decompose problem Extends mathematical programming with logical correlations 	<p>MINLP Reformulation</p> <ul style="list-style-type: none"> Assign a binary variable y for each logic variable Y Ensure that feasible regions are equivalent Big-M and Hull <p>Logic-based methods</p> <ul style="list-style-type: none"> Logic Branch-and-Bound (LBB) Logic Outer-Approximation (LOA) Activate and deactivate terms in disjunction Evaluate solutions and generate relaxations
Formulation	
$\min_{x,y,z} f(x,y,z)$ $s.t. g(x,y,z) \leq 0$ $\bigvee_{i \in D_k} [r_{ik}(x,y,z) \leq 0] \quad \forall k \in K$ $\Omega(Y) = \text{True}$ $x \in X \subseteq \mathbb{R}^{n_x}$ $Y \in \{\text{True, False}\}^{n_y}$ $z \in Z \subseteq \mathbb{Z}^{n_z}$	<p>Objective function Global constraints Disjunctions Logic correlations Continuous vars. Logic variables Discrete vars.</p>

Proposed modeling and solution tools

Chosen Platform  +  + 	MindtPy - Mixed-Integer Decomposition Toolbox in Pyomo <ul style="list-style-type: none"> Algorithms implementation: (Global) Outer-Approximation, Extended Cutting Planes Novel algorithmic enhancements: Feasibility pumps, regularization techniques, generalized McCormick cuts, single-tree implementation 	Pyomo.GDP <ul style="list-style-type: none"> Native modeling extension for GDP Automatic reformulation into MINLP
Tools	GDPOpt <ul style="list-style-type: none"> Algorithms implementation for GDP: LBB and (Global) LOA Enhanced with satisfiability solvers such as Z3 	
CORAMIN <ul style="list-style-type: none"> Refinable relaxation generator for nonlinear functions in MINLP and GDP Complementary with solvers in Pyomo, such as MindtPy and GDPOpt 		

GDPLib – A Library of GDP Problems

Library of GDP problems ready to be solved

- More than 25 different GDP problems relevant to **Process Systems Engineering** and **Superstructure Optimization**
 - Methanol production process
 - Hydrodealkylation (HDA) process to produce Toluene
 - Biofuel processing network
 - Heat exchanger network evaluating modular process design
 - Plant capacity expansion model
 - Synthesis gas production plant from methane
 - Kaibel distillation column
 - Tray distillation column design
- Problem ranging from 6 to 31968 continuous variables, 2 to 516 disjunctions, 0 to 5040 integer variables, and 30 to 14927 constraints
- `pip install gdplib`

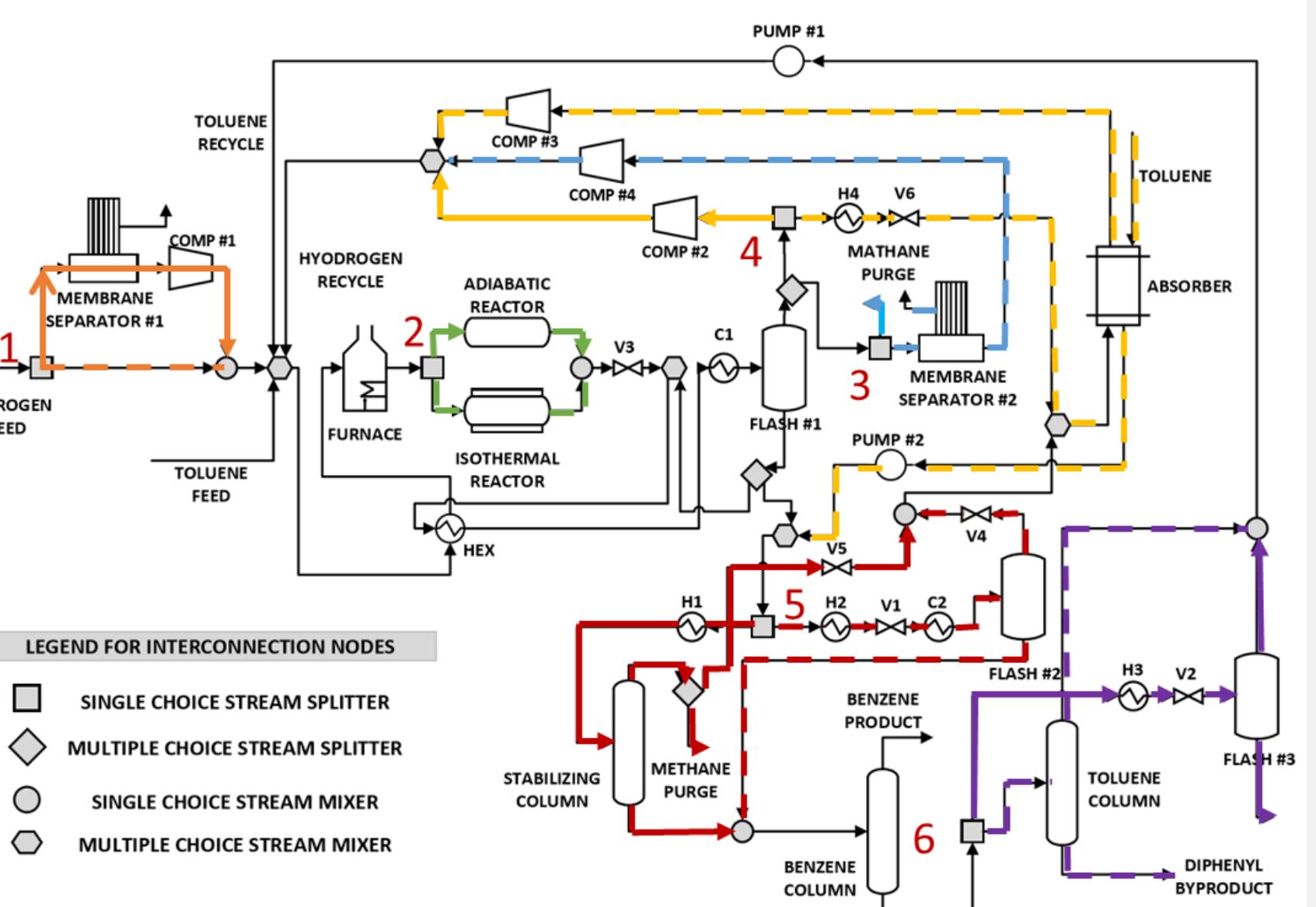
Example 1 – Methanol Production Process

Problem with 6 different disjunctions:

Hydrogen feed pretreatment, adiabatic or isothermal reactor, methane stream purge or recycle, absorber installation or vapor stream recycle, stabilizing column or a flash to remove extra methane, and Dyphenyl flash separation or Toluene distillation

- 285 variables
- 277 constraints
- 4 disjunctions
- $2^6 = 64$ different process alternatives
- Commercial global solvers fail to solve MINLP reformulation
- Complete and stable process alternatives enumeration using superstructure model
- LOA from GDPOpt is able to find global optimal solution in less than a minute in a normal desktop using free-access solvers

733 variables • 721 constraints • 6 disjunctions



LEGEND FOR INTERCONNECTION NODES

- SINGLE CHOICE STREAM SPLITTER
- MULTIPLE CHOICE STREAM SPLITTER
- SINGLE CHOICE STREAM MIXER
- MULTIPLE CHOICE STREAM MIXER

Feed 1 (cheap) → 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9 → 10 → 11 → 12 → 13 → Byp. (Byp. = Byp. of 10)

Feed 2 (expensive) → 14 → 15 → 16 → 17 → 18 → 19 → 20 → 21 → 22 → 23 → 24 → 25 → 26 → 27 → 28 → 29 → 30 → 31 → 32 → 33 → 34 → 35 → 36 → 37 → 38 → 39 → 40 → 41 → 42 → 43 → 44 → 45 → 46 → 47 → 48 → 49 → 50 → 51 → 52 → 53 → 54 → 55 → 56 → 57 → 58 → 59 → 60 → 61 → 62 → 63 → 64 → 65 → 66 → 67 → 68 → 69 → 70 → 71 → 72 → 73 → 74 → 75 → 76 → 77 → 78 → 79 → 80 → 81 → 82 → 83 → 84 → 85 → 86 → 87 → 88 → 89 → 90 → 91 → 92 → 93 → 94 → 95 → 96 → 97 → 98 → 99 → 100 → 101 → 102 → 103 → 104 → 105 → 106 → 107 → 108 → 109 → 110 → 111 → 112 → 113 → 114 → 115 → 116 → 117 → 118 → 119 → 120 → 121 → 122 → 123 → 124 → 125 → 126 → 127 → 128 → 129 → 130 → 131 → 132 → 133 → 134 → 135 → 136 → 137 → 138 → 139 → 140 → 141 → 142 → 143 → 144 → 145 → 146 → 147 → 148 → 149 → 150 → 151 → 152 → 153 → 154 → 155 → 156 → 157 → 158 → 159 → 160 → 161 → 162 → 163 → 164 → 165 → 166 → 167 → 168 → 169 → 170 → 171 → 172 → 173 → 174 → 175 → 176 → 177 → 178 → 179 → 180 → 181 → 182 → 183 → 184 → 185 → 186 → 187 → 188 → 189 → 190 → 191 → 192 → 193 → 194 → 195 → 196 → 197 → 198 → 199 → 200 → 201 → 202 → 203 → 204 → 205 → 206 → 207 → 208 → 209 → 210 → 211 → 212 → 213 → 214 → 215 → 216 → 217 → 218 → 219 → 220 → 221 → 222 → 223 → 224 → 225 → 226 → 227 → 228 → 229 → 230 → 231 → 232 → 233 → 234 → 235 → 236 → 237 → 238 → 239 → 240 → 241 → 242 → 243 → 244 → 245 → 246 → 247 → 248 → 249 → 250 → 251 → 252 → 253 → 254 → 255 → 256 → 257 → 258 → 259 → 260 → 261 → 262 → 263 → 264 → 265 → 266 → 267 → 268 → 269 → 270 → 271 → 272 → 273 → 274 → 275 → 276 → 277 → 278 → 279 → 280 → 281 → 282 → 283 → 284 → 285 → 286 → 287 → 288 → 289 → 290 → 291 → 292 → 293 → 294 → 295 → 296 → 297 → 298 → 299 → 300 → 301 → 302 → 303 → 304 → 305 → 306 → 307 → 308 → 309 → 310 → 311 → 312 → 313 → 314 → 315 → 316 → 317 → 318 → 319 → 320 → 321 → 322 → 323 → 324 → 325 → 326 → 327 → 328 → 329 → 330 → 331 → 332 → 333 → 334 → 335 → 336 → 337 → 338 → 339 → 340 → 341 → 342 → 343 → 344 → 345 → 346 → 347 → 348 → 349 → 350 → 351 → 352 → 353 → 354 → 355 → 356 → 357 → 358 → 359 → 360 → 361 → 362 → 363 → 364 → 365 → 366 → 367 → 368 → 369 → 370 → 371 → 372 → 373 → 374 → 375 → 376 → 377 → 378 → 379 → 380 → 381 → 382 → 383 → 384 → 385 → 386 → 387 → 388 → 389 → 390 → 391 → 392 → 393 → 394 → 395 → 396 → 397 → 398 → 399 → 400 → 401 → 402 → 403 → 404 → 405 → 406 → 407 → 408 → 409 → 410 → 411 → 412 → 413 → 414 → 415 → 416 → 417 → 418 → 419 → 420 → 421 → 422 → 423 → 424 → 425 → 426 → 427 → 428 → 429 → 430 → 431 → 432 → 433 → 434 → 435 → 436 → 437 → 438 → 439 → 440 → 441 → 442 → 443 → 444 → 445 → 446 → 447 → 448 → 449 → 450 → 451 → 452 → 453 → 454 → 455 → 456 → 457 → 458 → 459 → 460 → 461 → 462 → 463 → 464 → 465 → 466 → 467 → 468 → 469 → 470 → 471 → 472 → 473 → 474 → 475 → 476 → 477 → 478 → 479 → 480 → 481 → 482 → 483 → 484 → 485 → 486 → 487 → 488 → 489 → 490 → 491 → 492 → 493 → 494 → 495 → 496 → 497 → 498 → 499 → 500 → 501 → 502 → 503 → 504 → 505 → 506 → 507 → 508 → 509 → 510 → 511 → 512 → 513 → 514 → 515 → 516 → 517 → 518 → 519 → 520 → 521 → 522 → 523 → 524 → 525 → 526 → 527 → 528 → 529 → 530 → 531 → 532 → 533 → 534 → 535 → 536 → 537 → 538 → 539 → 540 → 541 → 542 → 543 → 544 → 545 → 546 → 547 → 548 → 549 → 550 → 551 → 552 → 553 → 554 → 555 → 556 → 557 → 558 → 559 → 560 → 561 → 562 → 563 → 564 → 565 → 566 → 567 → 568 → 569 → 570 → 571 → 572 → 573 → 574 → 575 → 576 → 577 → 578 → 579 → 580 → 581 → 582 → 583 → 584 → 585 → 586 → 587 → 588 → 589 → 589 → 590 → 591 → 592 → 593 → 594 → 595 → 596 → 597 → 598 → 599 → 600 → 601 → 602 → 603 → 604 → 605 → 606 → 607 → 608 → 609 → 610 → 611 → 612 → 613 → 614 → 615 → 616 → 617 → 618 → 619 → 620 → 621 → 622 → 623 → 624 → 625 → 626 → 627 → 628 → 629 → 630 → 631 → 632 → 633 → 634 → 635 → 636 → 637 → 638 → 639 → 640 → 641 → 642 → 643 → 644 → 645 → 646 → 647 → 648 → 649 → 650 → 651 → 652 → 653 → 654 → 655 → 656 → 657 → 658 → 659 → 660 → 661 → 662 → 663 → 664 → 665 → 666 → 667 → 668 → 669 → 670 → 671 → 672 → 673 → 674 → 675 → 676 → 677 → 678 → 679 → 680 → 681 → 682 → 683 → 684 → 685 → 686 → 687 → 688 → 689 → 689 → 690 → 691 → 692 → 693 → 694 → 695 → 696 → 697 → 698 → 699 → 700 → 701 → 702 → 703 → 704 → 705 → 706 → 707 → 708 → 709 → 709 → 710 → 711 → 712 → 713 → 714 → 715 → 716 → 717 → 718 → 719 → 719 → 720 → 721 → 722 → 723 → 724 → 725 → 726 → 727 → 728 → 729 → 729 → 730 → 731 → 732 → 733 → 734 → 735 → 736 → 737 → 738 → 739 → 739 → 740 → 741 → 742 → 743 → 744 → 745 → 746 → 747 → 748 → 749 → 749 → 750 → 751 → 752 → 753 → 754 → 755 → 756 → 757 → 758 → 759 → 759 → 760 → 761 → 762 → 763 → 764 → 765 → 766 → 767 → 768 → 769 → 769 → 770 → 771 → 772 → 773 → 774 → 775 → 776 → 777 → 778 → 779 → 779 → 780 → 781 → 782 → 783 → 784 → 785 → 786 → 787 → 788 → 789 → 789 → 790 → 791 → 792 → 793 → 794 → 795 → 796 → 797 → 798 → 799 → 799 → 800 → 801 → 802 → 803 → 804 → 805 → 806 → 807 → 808 → 809 → 809 → 810 → 811 → 812 → 813 → 814 → 815 → 816 → 817 → 818 → 819 → 819 → 820 → 821 → 822 → 823 → 824 → 825 → 826 → 827 → 828 → 829 → 829 → 830 → 831 → 832 → 833 → 834 → 835 → 836 → 837 → 838 → 839 → 839 → 840 → 841 → 842 → 843 → 844 → 845 → 846 → 847 → 848 → 849 → 849 → 850 → 851 → 852 → 853 → 854 → 855